

Verbesserung der Kontexterkenkung in Ubiquitären Informationsumgebungen

Von der Carl-Friedrich-Gauß Fakultät der
Technischen Universität Carolo-Wilhelmina zu Braunschweig genehmigte
D i s s e r t a t i o n
zur Erlangung des Grades eines
Doktors der Ingenieurwissenschaften (Dr.-Ing.)

von:	Dipl. Inform. Tobias H. Zimmer
Referent:	Prof. Dr. M. Beigl, Institut für Betriebssysteme und Rechnerverbund, TU Braunschweig
Korreferent:	Prof. Dr. W. Juling, Institut für Telematik, Universität Karlsruhe (TH)
Datum der Promotion:	19. Juni 2007
Eingereicht am:	17. April 2007

Danksagung

Die vorliegende Dissertation entstand im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Telematik der Universität Karlsruhe (TH). Im Januar 2002 trat ich dort in die Forschungsgruppe des Telecooperation Office (TecO) ein, die von Prof. Dr.-Ing. Michael Beigl geleitet wurde. Das TecO war mir bereits aus meiner früheren Tätigkeit als Hilfwissenschaftler und aus der Zeit der Anfertigung meiner Diplomarbeit bekannt. In dieser Zeit wurde mein Interesse für die Forschung im Bereich Ubiquitous Computing geweckt. Durch die flachen Hierarchien und die Möglichkeit zu selbstverantwortlicher Arbeit während meiner Zeit als Student und insbesondere während meiner Tätigkeit als wissenschaftlicher Mitarbeiter, wurde eine besonders angenehme Arbeitsatmosphäre geschaffen. Der tägliche Austausch mit den Mitarbeitern und Studenten, sowie das breite Interesse an vielen Forschungsfeldern hat eine kreative, motivierende und stimulierende Umgebung erzeugt, in der das Arbeiten und Forschen viel Spaß gemacht hat. Das TecO hat mir die Möglichkeit eröffnet, während meiner Zeit als Mitarbeiter eine Vielzahl wissenschaftlicher Konferenzen und Partnerinstitute in Deutschland und im Ausland zu besuchen und so für den intellektuellen Austausch unterstützt, der für die Forschung an aktuellen Themen notwendig ist. Mein besonderer Dank gilt Prof. Dr. W. Juling, der als Leiter des TecO gerne an unseren Vortragsreihen teilgenommen hat und mir bei meiner Promotion nicht nur mit interessanten Anregungen und nützlicher fachlicher Kritik zur Seite stand, sondern sich auch bereit erklärte, meine Arbeit als Korreferent zu betreuen. Neben den fachlichen Diskussionen hat mir seine langjährige Erfahrung sehr geholfen, meine zentralen wissenschaftlichen Beiträge klar herauszustellen. Weiterhin danke ich Prof. Dr. rer. nat. H. Hartenstein, der mit seiner fachlichen Kritik ebenfalls einen relevanten Beitrag zum Gelingen dieser Arbeit geleistet hat. Darüber hinaus möchte ich Prof. em. Dr. Dr.-Ing. Eh Dr. hc. mult. Gerhard Krüger dafür danken, dass er immer wieder Zeit für Besuche im TecO fand, und besonders die Strukturierung und Argumentation meiner Arbeit in die richtigen Wege leitete. Den Mitarbeitern des TecO bin ich ebenso zu besonderem Dank verpflichtet. Veröffentlichungen mit Albert Krohn und Christian Decker und viele gemeinsame Diskussionen, sowie der Austausch über weit gefächerte Themen, waren inspirierend und haben mir dabei geholfen, meine Ideen zu formulieren und zu kon-

kretisieren. Vor allem möchte ich Prof. Dr.-Ing. Michael Beigl, als Leiter des TecO, für seine Unterstützung danken. Er hat es verstanden, mich während meiner Tätigkeit stets zu motivieren und für neue Fragestellungen zu begeistern. Damit hat er einen außerordentlichen Beitrag zum Gelingen meiner Arbeit geleistet. Er hat sich stets für meine Ideen und Vorschläge interessiert und war ein wichtiger Diskussionspartner. Unter seiner professionellen Betreuung konnte ich meine wissenschaftlichen Beiträge in anerkannten Fachpublikationen veröffentlichen und so meine Position in der internationalen Forschung einnehmen. Er hat mich immer wieder ermutigt, meine Ideen konsequent zu Ende zu denken und meine neuen Beiträge im wissenschaftlichen Kontext, systematisch zu überprüfen und verständlich darzustellen.

Weiterhin gilt mein Dank Martin Berchtold und Pablo Esteves, die durch ihre überdurchschnittlichen Leistungen als Hilfwissenschaftler am TecO, die Implementierung und Evaluierung meiner Forschungsergebnisse ermöglicht haben.

Abschließend möchte ich meiner Familie danken: meiner Mutter, die mich durch die Zeit meiner Ausbildung an Schule und Universität stets unterstützte und meiner Partnerin, die mir während der Zeit meiner Promotion viel Rückhalt gegeben hat.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Ubiquitous Computing	1
1.2	Motivation	3
1.2.1	Motivierendes Beispiel und Problemstellung	3
1.2.2	Lösungsansatz	6
1.3	These und Ziele	7
1.4	Aufbau und Inhalt	8
2	Grundlagen	11
2.1	Abgrenzung und Einordnung der Themengebiete	11
2.1.1	Ubiquitous Computing: Kontext	11
2.1.2	Überblick der verwandten Themengebiete	12
2.1.3	Anwendungsbereiche dieser Arbeit	15
2.1.4	Eigenschaften der Anwendungsdomäne	16
2.2	Kontextbegriffe	18
2.2.1	Kontext	18
2.2.2	Kontextsensitivität	20
2.2.3	Architektur	21
2.3	Kontextmodellierung	23
2.3.1	Semantische Kontextmodelle	24
2.3.2	Funktionale Kontextmodelle	27
2.4	Kontextkommunikation	29
2.4.1	Mobile Ad-hoc-Netzwerke	29
2.4.2	Relevante Aspekte für diese Arbeit	30
2.5	Kontextverarbeitung	30
2.5.1	Klassifikation und Aggregation	31
2.5.2	Fusion	31
2.5.3	Relevante Aspekte für diese Arbeit	31
2.6	Kontextmanagement	32
2.6.1	Qualität der Kontexterkenennung	32
2.7	Zusammenfassung	36

3	Modell	38
3.1	Thesen und Modelle	39
3.2	Kostenmodell der Kontextkommunikation und Kontextverarbeitung	40
3.2.1	Untersuchte Kommunikations- und Verarbeitungsmodelle	41
3.2.2	Ergebnisse der Kostenrechnung	42
3.3	Modell der Kontextkommunikation und Kontextverarbeitung . . .	43
3.3.1	Anforderungen an Kommunikation und Kontextverarbeitung in ubiquitären Informationsumgebungen	43
3.4	Kontext als Datenstruktur	46
3.4.1	Einordnung und Begriffsbildung	47
3.4.2	Aufbau eines Kontextes	51
3.5	Produzent-Konsument-Schema	54
3.6	Kontexthierarchie	56
3.6.1	Dynamisches Verarbeitungsmodell für Kontexte	56
3.7	Entkoppelte Kommunikation: Kontext-Raum	58
3.7.1	Operatoren des ContextSpace	60
3.7.2	Einordnung des CSpace bezüglich des Kostenmodells . . .	62
3.8	Attributierte Kontexte	63
3.8.1	Dimensionen semantischer Kontexte	64
3.8.2	Temporale Dimension	66
3.8.3	Räumliche Dimension	69
3.8.4	Verlässlichkeit von Kontext als Dimension des Informationsraumes	71
3.8.5	Abhängigkeit zwischen Kontexten als Dimension des Informationsraumes	76
3.8.6	Zusammenfassung	82
3.9	Zusammenfassung	83
4	Architektur	85
4.1	Architektur des Artefaktes und der ubiquitären Informationsumgebung	86
4.2	AwareCon und ConCom: Kontextkommunikation und Kontext-Raum	88
4.2.1	Realisierung des CSpace-Modells im Kommunikationskanal	90
4.3	Architektur des Kontextmanagementsystems	91
4.3.1	Empfang eines Kontextes	93
4.3.2	Versand eines Kontextes	94

4.4	Aufbau der Kontextattribute	94
4.5	Kontextalter (CA)	96
4.5.1	Setzen des Kontextalters	97
4.5.2	Auswertung des Kontextalters	99
4.6	Räumliche Herkunft (LOC)	100
4.6.1	Lokationsmodell des RAUM-Systems	100
4.6.2	Setzen der räumlichen Herkunft	101
4.6.3	Auswertung der räumlichen Herkunft	103
4.7	Verlässlichkeit von Kontexten (FCR)	104
4.7.1	Verlässlichkeit in unscharfer Logik	104
4.7.2	Formalisierung: Generierung und Anpassung des Fuzzy- Systems während der Entwicklungsphase	108
4.7.3	Berechnung und Setzen der Verlässlichkeit	116
4.7.4	Auswertung der Verlässlichkeit	117
4.8	Verwandschaft von Kontexten (GRC)	118
4.8.1	Genome von Kontexten	119
4.8.2	Probabilistic Multi-Site Crossover (PMSC)	121
4.8.3	Berechnung des Verwandtschaftsgrades von Kontexten	125
4.8.4	Systematische Überschätzung des Verwandtschaftsgrades	128
4.8.5	Setzen des Kontextgenoms	135
4.8.6	Auswertung des Kontextgenoms	136
4.9	Der Qualitätsfilter	137
4.9.1	Filterung: CA	138
4.9.2	Filterung: LOG	138
4.9.3	Filterung: FCR	138
4.9.4	Filterung: GRC	139
4.10	Zusammenfassung	140
5	Implementierung	142
5.1	Rahmenbedingungen für die Umsetzung des Kontextmanagement- systems	142
5.1.1	Reale Anwendungsdomäne	143
5.1.2	Simulation	144
5.2	Simulation von ubiquitären Informationsumgebungen	145
5.2.1	Verfügbare Simulatoren	145
5.2.2	Simulator: context_sim	148
5.2.3	Qualitätsmanagementschicht: QoC	150
5.3	Reale Artefakte: AwareOffice	151

5.3.1	AwarePen	152
5.3.2	AwareSponge	154
5.3.3	AwareCam	155
5.3.4	Stühle, Fenster, Tür, Projektor, Weißwandtafel	156
5.3.5	Interaktives Türschild	157
5.3.6	Kontextsensitive PDAs	158
5.4	Zusammenfassung	159
6	Evaluierung	161
6.1	Simulation: GRC in MATLAB	161
6.1.1	Adjustierbare Parameter von GRC	162
6.1.2	Auswirkung der Parameter Variation	162
6.2	Simulation: context_sim	165
6.2.1	Attribut CA: Kontextalter	167
6.2.2	Attribut LOC: Lokation	174
6.2.3	Attribut FCR: Verlässlichkeit	177
6.2.4	Attribut GRC: Verwandtschaft	181
6.2.5	Gesamtszenario	188
6.3	Feldversuch: AwareOffice	193
6.3.1	Feldversuch: CA und LOC	194
6.3.2	Feldversuch: FCR	196
6.3.3	Feldversuch: GRC	198
6.4	Zusammenfassung	200
7	Zusammenfassung und Ausblick	201
7.1	Zusammenfassung der Ergebnisse	201
7.2	Ausblick	203
	Literaturverzeichnis	205
	Abbildungsverzeichnis	224
	Tabellenverzeichnis	227
	Glossar	228
	Anhang	

A	Simulationsergebnisse	233
A.1	Simulation: CA	233
A.1.1	CPA ohne Filter	233
A.1.2	CPA mit Filter	237
A.1.3	CSA ohne Filter	240
A.1.4	CSA mit Filter 444	242
A.1.5	CSA mit Filter 234	244
A.2	Simulation: LOC	245
A.2.1	LOC ohne Filter	245
A.2.2	LOC mit Filter	247
A.3	Simulation: FCR	249
A.3.1	FCR ohne Filter	249
A.3.2	FCR mit Filter	251
A.4	Simulation: GRC	253
A.4.1	GRC Auffächerung und Multiplikation ohne Filter	253
A.4.2	GRC Auffächerung und Multiplikation mit Filter	255
A.4.3	GRC zyklische Verarbeitung ohne Filter	257
A.4.4	GRC zyklische Verarbeitung mit Filter	259
A.5	Simulation: Gesamtszenario	261
A.5.1	Zusammenfassung der Simulationsergebnisse ohne Filter	261
A.5.2	Zusammenfassung der Simulationsergebnisse mit Filter	261
A.5.3	Simulationsgraph	263
B	Studie: Zählung der Gegenstände in Büroumgebungen	264
B.1	Bürozählung: A. Krohn	265
B.2	Bürozählung: P. Spieß	275
B.3	Bürozählung: T. Zimmer	280
B.4	Bürozählung: C. Decker	289

1 Einleitung

Thema der vorliegenden Arbeit ist die Verarbeitung von Kontexten in ubiquitären Informationsumgebungen. Ziel dieser Arbeit ist es, Systeme, die Kontexte verarbeiten, mit neuen Verfahren systematisch zu erweitern, so dass eine signifikante Verbesserung in der Erkennung von Kontexten erreicht werden kann. Dazu wird das Konzept des systematischen Kontextmanagements eingeführt, das eine neue Schicht innerhalb kontextverarbeitender Systeme bildet. Das Konzept des Kontextmanagements wird in dieser Abhandlung nicht nur theoretisch erarbeitet, sondern auch praktisch umgesetzt und anhand von Simulationen und implementierten Anwendungen evaluiert.

In dieser Einleitung wird zunächst der Begriff des „Ubiquitous Computing“ eingeführt und die Notwendigkeit der Betrachtung eines systematischen Kontextmanagements motiviert. Darauf aufbauend werden die Ziele und Thesen dieser Arbeit formuliert. Die Einleitung schließt mit einer Übersicht über die angewandte Methodik und den Aufbau und Inhalt der Arbeit.

1.1 Ubiquitous Computing

Der Begriff „Ubiquitous Computing“ geht auf Mark Weiser zurück, der ihn im Jahre 1991 in einem Artikel mit dem Titel „*The computer for the 21st century*“ im Scientific American einführte [176]. Zu dieser Zeit entstanden unter Weisers Leitung die ersten Ubiquitous Computing Systeme im Xerox-Parc Forschungslabor in Palo Alto, das ParcTab [173] und das Reactive Environment [43][26]. Im deutschen Sprachraum wird vorwiegend der englische Begriff Ubiquitous Computing als Eigenname des Forschungsbereiches verwendet, die Übersetzung „ubiquitäres Rechnen“ ist eher selten zu finden. Auch die vorliegende Arbeit folgt diesem Sprachgebrauch, benutzt aber ansonsten die deutschen Bezeichnungen.

Ubiquitous Computing beschreibt nach Weiser vier prägnante Entwicklungen in der Benutzung und Verbreitung computerisierter Systeme:

- Computer werden unsichtbar. Sie verschmelzen mit der Umgebung so, dass der Benutzer den Computer nicht mehr als solchen wahrnimmt. Er wird integraler Bestandteil eines Gegenstandes oder einer Umgebung.
- Computer werden ubiquitär verfügbar. Sinkende Größe, steigende Leistungsfähigkeit und sinkender Preis von Computern führt zu einer rapide steigenden Verbreitung.
- Computer können ubiquitär kommunizieren. Die Entwicklung auf dem Sektor der funkbasierten Kommunikation ermöglicht eine weitgehende Vernetzung der Computer.
- Die Beziehung zwischen Mensch und Computer erfährt den zweiten grundlegenden Wandel. Ein Benutzer besitzt und interagiert nicht mehr nur mit einem Gerät, sondern mit einer Vielzahl von Computern.

Der signifikanteste Wandel den Weiser sieht, ist ein Paradigmenwechsel in der Interaktion mit Computern und dem Verständnis von Computern. In seinem Artikel „*Designing Calm Technology*“ [177] beschreibt er Ubiquitous Computing als dritte Ära der Computerbenutzung nach der Mainframe Ära und nach der Ära des Personalcomputers. Zu Zeiten der Mainframebenutzung teilten sich viele Nutzer die Ressourcen eines einzigen Rechners. Der Benutzer musste ein großes Maß an Fachwissen mitbringen, um in der Lage zu sein, den Computer benutzen zu können. Dies änderte sich grundlegend mit dem Aufkommen der Personalcomputer ab 1980. Weiser und seine Kollegen sahen bereits in den 1990er Jahren den nächsten anstehenden Wandel voraus: weg von dem personalisierten Computer hin zu einer Umgebung, die mit computerisierten Gegenständen, Computern und Kommunikationselektronik durchsetzt ist.

Heute bezeichnet Ubiquitous Computing (kurz: UbiComp) einen Forschungsbereich, der sich in mehrere selbständige Teilgebiete untergliedern lässt, in denen sich Wissenschaftler mit unterschiedlichen Fragestellungen beschäftigen. Zentrale Bereiche sind hier mobile drahtlose Ad-hoc-Netzwerke, Mensch-Maschine Interaktion und Benutzerschnittstellen, mobile verteilte Sensornetzwerke, Systemsoftware, Betriebssysteme und Kontextverarbeitung.

1.2 Motivation

Ubiquitäre Informationsumgebungen existieren heute zumeist nur als geschlossene Systeme in Laboren. Im Unterschied zu der von Mark Weiser formulierten Vision zukünftiger Lebens- und Arbeitsumgebungen, bestehen diese Laborumgebungen aus einer festen Auswahl von Geräten und oft nur einer oder weniger Anwendungen, die dann im Zentrum des wissenschaftlichen Interesses stehen.

Ubicomp als technologisches und gesellschaftliches Phänomen zeichnet dagegen ein Bild offener Bereiche, in denen sich ubiquitäre Informationsumgebungen ad hoc ausbilden können, wann immer genügend mobile computerisierte Geräte zusammenkommen, so dass sie in der Lage sind zu kommunizieren und mit den ausgetauschten (Kontext-) Informationen durch ihre Anwendungen einen Mehrwert für den Benutzer, der sich in einer solchen Umgebung bewegt, zu schaffen. Dabei identifizieren viele Autoren das Potential solcher neuer Strukturen in den Möglichkeiten, die sich aus der dynamischen Komposition von Diensten und Dienstleistungen durch die Vielzahl mobiler computerisierter Gegenstände ergeben.

Heute steht die Forschung an der Schwelle, an der die ersten Geräte und Anwendungen bereit sind, die Labore zu verlassen. Damit stellen sich aber auch neue Fragen und Probleme, deren Betrachtung bis jetzt nicht im zentralen Fokus der Forschungsarbeit stand. Das wichtigste dieser Probleme, die auf dem Weg zu ubiquitärer unaufdringlicher und doch nachhaltiger Computerunterstützung in alltäglichen Umgebungen gelöst werden muss, ist das der Komplexität dieser Szenarien. Zu der Komplexität ubiquitärer Informationsumgebungen trägt neben der hohen Anzahl von Geräten und deren potentieller Mobilität, auch die dynamische Verknüpfung ihrer Anwendungen bei.

1.2.1 Motivierendes Beispiel und Problemstellung

In zukünftigen, realen ubiquitären Informationsumgebungen¹ werden Dienste für die Benutzer erbracht werden, indem die Anwendungen in diesen Umgebungen kooperieren und Daten in Form von Kontexten austauschen. Dabei werden sowohl die Anzahl als auch die Art der in einer Umgebung verfügbaren Geräte nicht

¹Der Begriff „reale ubiquitäre Informationsumgebungen“ wird hier in Abgrenzung von überwachten Laborumgebungen verwendet.

vorgegeben sein, sondern durch die Dynamik und Mobilität in der Umgebung bestimmt werden.

Diese Grundbedingungen führen zu komplexen Verarbeitungsstrukturen, die innerhalb der ubiquitären Informationsumgebung gehandhabt werden müssen. Zum einen diktiert sie Anforderungen an die Architektur der Kommunikation und Kontextverarbeitung in diesen Umgebungen, zum anderen beeinflussen sie die Qualität der Dienste, die von Anwendungen kooperativ erbracht werden. Als motivierendes Beispiel für diese Problematik soll das AwareOffice [164, 190] dienen.

Das AwareOffice ist eine Büroumgebung, die durch eingebettete Ubiquitous Computing Technologie in der Lage ist, ihre Benutzer bei typischen Aufgaben im Büroumfeld aktiv zu unterstützen. Gegenstände, in die Computer eingebettet wurden, umfassen unter anderem Stühle, Tassen [16] und Stifte (in diesem Fall Boardmarker zum Schreiben auf Weißwandtafeln). Hinzu kommen persönliche digitale Assistenten (PDAs), ein interaktives Türschild und eine Digitalkamera.

Basierend auf diesen Geräten wurden Anwendungen entwickelt, welche in der AwareOffice Umgebung verschiedene Dienste bereitstellen können. Zur Illustration stellt Abbildung 1.1 diesen kleinen Ausschnitt des AwareOffice schematisch dar: Geräte werden als Quadrate dargestellt und Kontexte, welche diese Geräte austauschen können, werden durch Ellipsen repräsentiert.

In einer realen ubiquitären Informationsumgebung ist damit zu rechnen, dass viele der aktiven Gegenstände von verschiedenen Herstellern stammen werden. Diese Heterogenität führt dazu, dass die Kommunikation und Datenaustauschformate standardisiert werden müssen, um eine Zusammenarbeit zu ermöglichen. Hinzu kommt, dass die Verarbeitung der Kontexte in Geräten verschiedener Hersteller auf unterschiedlichen Algorithmen beruhen wird. Dies führt in der Konsequenz zu unterschiedlichen Qualitäten der Ausgaben dieser Geräte.

Darüber hinaus werden in realen ubiquitären Informationsumgebungen mehrere Gegenstände eines Typs zur gleichen Zeit präsent sein, wobei, wie zuvor erwähnt, die Eigenschaften der von ihnen bereitgestellten Daten durchaus unterschiedliche sein können. Abbildung 1.2 zeigt die Darstellung einer AwareOffice Umgebung, in der die gleichen Geräteklassen wie in Abbildung 1.1 präsent sind. Allerdings wurden hier die Kontexte verschiedener Geräte unterschieden, da diese unterschiedliche Eigenschaften haben können und so auch für den Empfänger in einer ubiquitären Informationsumgebung unterscheidbar sein sollten.

Der Graph zeigt die Datenverarbeitung von 5 Tassen, 6 Stühlen, 2 Stiften, 5 PDAs, 1 Türschild und 1 Kamera, die diese AwareOffice Umgebung bilden, al-

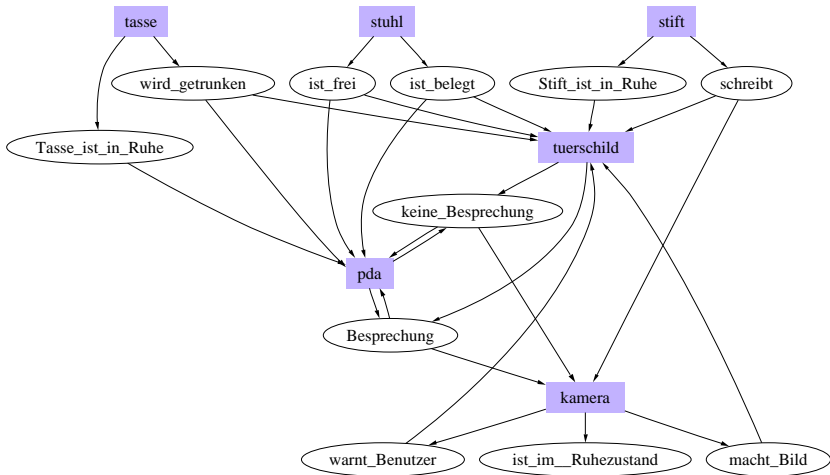


Abbildung 1.1. Reduzierter Verarbeitungsgraph einer ubiquitären Informationsumgebung

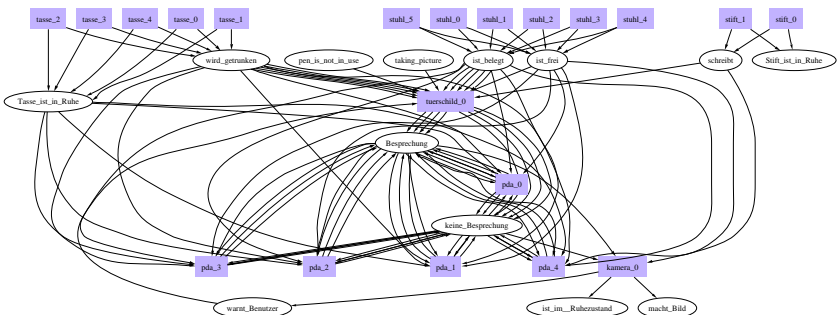


Abbildung 1.2. Vollständiger Verarbeitungsgraph

so insgesamt von nur 18 Geräten und deren Anwendungen. Selbst diese geringe Anzahl von aktiven Geräten in einer Umgebung führt, wie zu sehen ist, bereits zu komplexen Zusammenhängen und Abhängigkeiten. Legt man nun ubiquitäre Informationsumgebungen zu Grunde, wie sie in Ubicomp diskutiert werden, an denen mehrere hundert Geräte und Anwendungen beteiligt sind, lässt sich die Komplexität der entstehenden Strukturen extrapolieren.

Neben der systematischen Komplexität großer ubiquitärer Informationsumgebungen, ist die Komplexität der Kontextverarbeitung zu beachten. Die mehrstufige Kontextverarbeitung in diesen Umgebungen führt zu einer potentiellen Degression der Qualität der Kontextdaten: Fehler und Ungenauigkeiten in der Kontexterfassung pflanzen sich ohne geeignete Qualitätssicherungsmechanismen über die Verarbeitungskette der Kontexte fort und können sich bei der Ableitung von Kontexten aus mehreren Quellen sogar addieren.

Diese Komplexität im Datenaustausch und der Datenverarbeitung in Ubiquitous Computing Systemen handhabbar zu machen und dabei die Qualität der Kontexte zu sichern, ist eine der großen Herausforderungen der Forschung im Bereich ubiquitärer Informationstechnologien, zu deren Bewältigung diese Arbeit einen Beitrag leisten will.

1.2.2 Lösungsansatz

Diese Arbeit stellt eine Architekturweiterung für ubiquitäre Informationsumgebungen vor, welche zur Lösung der oben aufgezeigten Probleme beiträgt. Kommunikation und Datenrepräsentation und -verarbeitung in ubiquitären Informationsumgebungen müssen systematisiert werden. Dadurch können Geräte aus verschiedenen Quellen zusammenarbeiten. Die Systematisierung bildet dabei den ersten Schritt in Richtung einer Standardisierung von Verfahren und Protokollen für ubiquitäre Informationsumgebungen.

Außerdem muss ein Qualitätsmanagementsystem etabliert werden, welches es den Teilnehmern einer ubiquitären Informationsumgebung erlaubt, auch angesichts komplexer Verarbeitungszusammenhänge qualitativ hochwertige Dienste zu liefern. Die Ansätze hierfür ergeben sich zunächst aus empirischen Studien und den Erfahrungen aus der Entwicklung und dem Umgang mit ubiquitären Informationstechnologien.

1.3 These und Ziele

Große ubiquitäre Informationsumgebungen stellen besondere Anforderungen an die Effizienz von Verfahren zur Kontextrepräsentation, Kontextkommunikation und Kontextverarbeitung. Bei der Verarbeitung von Kontexten steht dabei die Qualitätssicherung der Kontexterkennung im Verarbeitungsprozess im Vordergrund, da sie die Basis für qualitativ hochwertige Dienste innerhalb der ubiquitären Informationsumgebung bildet.

Die vorliegende Arbeit stellt zur Lösung der in Abschnitt 1.2.1 angesprochenen Probleme neue Verfahren der Kontextrepräsentation, Kontextkommunikation und Kontextverarbeitung und insbesondere der Qualitätssicherung in der Kontextverarbeitung vor. Die Grundidee hinter dieser Methodik ist es, eine systematische Basis für ubiquitäre Informationsumgebungen zu schaffen, auf deren Grundlage sich effiziente Qualitätssicherungsverfahren umsetzen lassen. Dies führt zu der Hauptthese dieser Arbeit:

These (Kontexterkennung in ubiquitären Informationsumgebungen)

Der Betrieb von großen ubiquitären Informationsumgebungen erfordert die systematische Ausgestaltung von Kontextrepräsentation, -kommunikation und -verarbeitung. In diesen Umgebungen kann die Kontexterkennung verbessert werden, indem Qualitätsmanagementmechanismen zur Überwachung der Qualität von Kontexten eingesetzt werden.

Es ist das Ziel dieser Arbeit die ersten Schritte auf dem Weg zu großen heterogenen ubiquitären Informationsumgebungen zu gehen und die dafür notwendigen Verfahren bereitzustellen. Die Hauptthese dieser Arbeit kann unter diesem Gesichtspunkt zu zwei Subthesen spezialisiert werden. Im zentralen Fokus steht dabei die Verbesserung der Kontexterkennung in ubiquitären Informationsumgebungen:

Subthese 1 (Verbesserung der Qualität der Kontexterkennung)

Die Erkennung von Kontexten in ubiquitären Informationsumgebungen kann qualitativ verbessert werden, wenn bei der Interpretation Eigenschaften, sogenannte Attribute, von Kontextdaten berücksichtigt werden.

Als Grundlage für ein Qualitätsmanagementsystem für Kontexte müssen allerdings zuerst Rahmenbedingungen geschaffen werden, welche die effiziente Umsetzung eines solchen Systems auch in großen, heterogenen ubiquitären Informationsumgebungen erlauben:

Subthese 2 (Effiziente Gestaltung ubiquitärer Informationsumgebungen)

Ubiquitäre Informationsumgebungen einer Größe von mehreren hundert Geräten führen zu hoch komplexen Verarbeitungsstrukturen. Diese Umgebungen können effektiv und effizient gestaltet werden, wenn für die Repräsentation, Kommunikation und Verarbeitung von Kontexten auf Systemebene einheitliche, optimierte Verfahrensprinzipien zum Einsatz kommen.

Analog zu der Spezialisierung der Hauptthese, können die Ziele der Arbeit konkretisiert werden. Ziel der Arbeit ist es, in einem ersten Schritt das Modell einer großen heterogenen ubiquitären Informationsumgebung zu entwerfen und zu einer umsetzbaren Architektur zu konkretisieren. Dieses Modell beinhaltet die grundlegenden technischen Parameter eines Ubiquitous Computing System: Kontextrepräsentation, Kontextkommunikation und Kontextverarbeitung. Es dient zur Erreichung des Hauptzieles dieser Arbeit, nämlich der Modellierung und Umsetzung eines effizienten Qualitätsmanagements für Kontexte in ubiquitären Informationsumgebungen, das zu einer Verbesserung der Kontexterkenkung von Anwendungen führt und somit die Qualität von Diensten in Ubiquitous Computing-Systemen verbessert.

1.4 Aufbau und Inhalt

Zunächst wird in Kapitel 2 der Stand der Forschung im Bereich Kontextverarbeitung in Ubiquitous Computing Systemen diskutiert und die vorliegende Arbeit in diesen Forschungsbereich eingeordnet. Untersucht werden hier schwerpunktmäßig die Bildung des Kontextbegriffes in der Forschung, die Modellierung von Kontexten, sowie der aktuelle Stand bei der Kommunikation und Verarbeitung von Kontexten. Abschließend werden Entwicklungen im Bereich des Kontextmanagements betrachtet.

In Kapitel 3 wird das Modell der Kontextkommunikation und Kontextverarbeitung und das darauf aufbauende Modell der Qualitätssicherung von Kontexten durch Kontextattribute entwickelt. Die formale Beschreibung von Kontexten sowie deren Verarbeitung und Kommunikation ermöglicht die Ausnutzung allgemeiner Systemeigenschaften zum Aufbau von Qualitätssicherungsmechanismen. Die Qualität von Kontexten wird dabei unabhängig von der inhaltlichen Bedeutung der Kontextdaten für eine Anwendung bestimmt und ausgewertet. Das Modell der Kontextkommunikation und Kontextverarbeitung beruht auf einer An-

forderungsanalyse und einem Kostenmodell, welches zur Analyse und zum Vergleich der Effizienz verschiedener möglicher alternativer Modellierungen herangezogen wurde. Das Modell der Qualitätssicherung durch Kontextattribute wurde aus einer schrittweisen Problemanalyse heraus entwickelt, welche Einflussfaktoren auf die Kontextqualität identifiziert.

In Kapitel 4 werden die Modelle, welche im dritten Kapitel als effiziente Umsetzungen einer ubiquitären Informationsumgebung identifiziert wurden in einer Architektur konkretisiert. In diesem Kapitel werden die Verfahren zur Qualitätssicherung eingehend diskutiert und die Algorithmen eingeführt, welche die konkreten Qualitätsmaße berechnen.

Kapitel 5 beschreibt die Implementierung der zuvor eingeführten Architektur sowohl im Rahmen einer Simulationsumgebung für große ubiquitäre Informationsumgebungen, als auch deren Umsetzung in den realen Anwendungen der AwareOffice Umgebung.

Die Ergebnisse der Evaluierung des entwickelten Systems werden in Kapitel 6 zusammengefasst. Dabei wird die Verbesserung der Kontexterkennung in realen Laborumgebungen und simulierten Großszenarien quantitativ betrachtet.

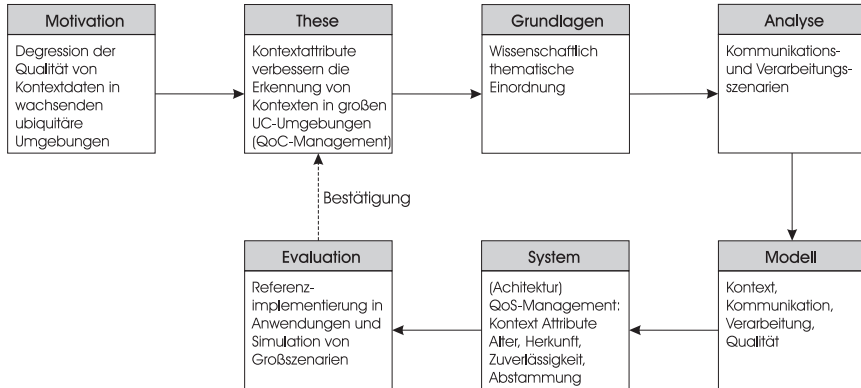


Abbildung 1.3. Aufbau dieser Arbeit

Abbildung 1.3 zeigt schematisch den methodischen Aufbau der Arbeit, welcher dem Inhalt der Kapitel weitgehend folgt: aus Motivation und Problemstellung wird die These der Arbeit abgeleitet. Zunächst werden dann die Grundlagen

der Arbeit erläutert und eine genaue thematische Einordnung in den Forschungsbereich Ubiquitous Computing vorgenommen. Daraufhin werden die Anforderungen an eine ubiquitäre Informationsumgebung mit Qualitätssicherungsmechanismen analysiert und basierend darauf die Modelle dieser Arbeit entwickelt.

Im nächsten Schritt werden diese Modelle in einer Systemarchitektur umgesetzt, welche die nötigen Verfahren und Algorithmen spezifiziert. Diese werden für die Evaluierung implementiert. Zur Bewertung der Verbesserung in der Kontexterkenennung dient eine Referenzimplementierung auf einer Hardwareplattform und Simulationsergebnisse aus der Simulation komplexer Szenarien, die sich heute aus Kostengründen noch nicht zu Testzwecken in Hardware realisieren lassen. Durch ein positives Ergebnis der Evaluierung wird die These dieser Arbeit bestätigt.

2 Grundlagen

Dieses Kapitel führt die Grundlagen der vorliegenden Ausarbeitung ein. Obwohl diese Arbeit einen völlig neuen Ansatz vorstellt, finden sich wichtige Fragestellungen, die hier behandelt werden, auch in Forschungsarbeiten Dritter. Im Folgenden werden einige dieser Arbeiten vorgestellt und zur Einordnung und Abgrenzung der vorliegenden Arbeit in den Forschungsbereich „Kontextverarbeitung in ubiquitären Informationsumgebungen“ herangezogen. Weiterhin soll dieses Kapitel wichtige Grundlagen, die zum besseren Verständnis dieser Arbeit beitragen, einführen. Dazu wird der Stand der Forschung in den Bereichen, die diese Arbeit tangieren erörtert. Im Einzelnen sind dies die Themen Kontextmodelle, Kontextkommunikation und Kontextverarbeitung.

2.1 Abgrenzung und Einordnung der Themengebiete

Die folgenden Abschnitte führen kurz in den Forschungsbereich des „Ubiquitous Computing“ ein und geben einen Überblick über die Themenbereiche, die diese Arbeit berühren. Die Forschungstätigkeit der vorliegenden Arbeit wird in den Gesamtthemenkomplex eingeordnet. Dabei wird der Anwendungsbereich der hier erzielten Forschungsergebnisse umrissen und die Eigenschaften der zu Grunde liegenden Anwendungsdomäne untersucht.

2.1.1 Ubiquitous Computing: Kontext

Die Grundidee des Ubiquitous Computing ist, dass Computer mehr und mehr in die Umgebung der Menschen und Gegenstände, mit denen sie arbeiten, eingebettet werden. Dort bieten sie dem Benutzer Dienste an, die ihn bei seinen täglichen Aufgaben unterstützen. Dabei bleibt der Computer selbst als Gerät im Hintergrund. Ziel der Entwicklung von Ubiquitous Computing Systemen ist es, die Dienstleistung möglichst unaufdringlich für den Benutzer zu gestalten. Dazu gehört, dass die Aufmerksamkeit des Benutzers nicht mehr als unbedingt notwendig belastet wird, was durch die konsequente Reduktion expliziter Interaktion

geschieht. An ihre Stelle tritt die implizite Interaktion mit dem Benutzer und der Umgebung. Integrierte Sensorik und eine Funkschnittstelle ermöglichen es Ubiquitous Computing Systemen, Informationen über ihre Umwelt zu sammeln und diese mit anderen Geräten zu kommunizieren. Die so zusammengetragenen Informationen werden benutzt, um Rückschlüsse auf die aktuelle Situation – den *Kontext* – zu ziehen, in der sich ein System befindet. Wird eine Situation oder ein Kontext erkannt, die für das System relevant ist, passt die Ubiquitous Computing Anwendung ihr Verhalten automatisch an. So wird ein explizites Eingreifen des Benutzers unnötig. Die Anwendungen einer ubiquitären Informationsumgebung passen ihr Verhalten dynamisch der Situation an.

Kontext bildet in ubiquitären Informationsumgebungen die Basis der Interaktion von Anwendungen mit dem Benutzer. Da einzelne Geräte und damit ihre Anwendungen nicht, oder nur in den seltensten Fällen in der Lage sind, alleine genug Daten zu erheben, um eine Situation korrekt zu beurteilen, werden die gesammelten Teilinformationen als Kontexte zwischen den Ubiquitous Computing Anwendungen kommuniziert und von diesen weiterverarbeitet.

Im Forschungsschwerpunkt Kontextverarbeitung in ubiquitären Informationsumgebungen lassen sich die Teilgebiete *Kontextmodellierung*, *Kontexterkennung*, *Kontextkommunikation*, *Kontextaggregation und -fusion*, und *Kontextmanagement* identifizieren.

2.1.2 Überblick der verwandten Themengebiete

Die vorliegende Arbeit ist im Bereich Kontextmanagement angesiedelt. Sie hat zum Ziel ein Qualitätsmanagementsystem für Kontexte in ubiquitären Informationsumgebungen bereitzustellen. Alle Bereiche der Forschung zum Themenkomplex Kontextverarbeitung werden durch diese Arbeit mehr oder weniger stark berührt, da sich ein Kontextmanagementsystem in die Strukturen der Kontextverarbeitung integrieren und mit den anderen Komponenten kontextverarbeitender Systeme interagieren muss.

Kontextmodellierung

Die Modellierung von Kontexten hat zum Ziel, die Informationen, die in einem Ubiquitous Computing System gewonnen, verarbeitet und kommuniziert werden,

zu strukturieren. Bei gängigen Modellierungsansätzen ist zwischen semantischen Kontextmodellen und funktionalen Kontextmodellen zu unterscheiden. Semantische Kontextmodelle modellieren in erster Linie die Abbildung der gesammelten Informationen auf abstrakte Situationen. Dabei steht die Interpretation der Kontexte im Vordergrund. Oft inkludieren semantische Kontextmodelle auch funktionale Bereiche der Kontextverarbeitung. Dies ist zumeist historisch bedingt, da sich die ersten wissenschaftlich formulierten Kontextmodelle sowohl mit semantischen, als auch funktionalen Aspekten auseinander setzten, allerdings dabei die semantischen Aspekte deutlich in den Vordergrund stellten [115, 141, 50].

Rein funktionale Kontextmodelle existieren bis heute praktisch nicht. Die in der Literatur beschriebenen funktionalen Modelle sind Teil monolithischer Modellierungen, die funktionale sowie semantische Aspekte der Kontextdarstellung beinhalten. Dies hat unter anderem auch zur Folge, dass heute, bis auf einzelne Ausnahmen, wie ConCom [106], kaum Systeme zur Verfügung stehen, die Kontext unabhängig von ihrer semantischen Modellierung betrachten.

Der Nachteil dieses Umstandes ist, dass heute praktisch keine interoperablen Ansätze in der Kontextverarbeitung existieren. Heutige Ubiquitous Computing Systeme sind damit auf die einmalige Auswahl eines semantischen Kontextmodells beschränkt, welches alle weiteren Systemkomponenten vorgibt.

An dieser Stelle tangiert die vorliegende Arbeit diesen Forschungsschwerpunkt, da hier ein Kontextmanagementsystem vorgestellt wird, das auf ein allgemeines, von semantischen Modellierungen unabhängiges funktionales Kontextmodell aufbaut.

Kontexterkenkung, Kontextaggregation und -fusion

Die beiden verwandten Forschungsbereiche Kontexterkenkung und Kontextaggregation und -fusion beschäftigen sich mit der Verarbeitung von Kontexten auf algorithmischer Ebene.

In der Kontexterkenkung werden Algorithmen untersucht, mit deren Hilfe auf der Basis von Sensordaten und eventuell weiteren Informationen über die Umgebung, Rückschlüsse auf die aktuelle Situation gezogen werden können. Hier beschäftigt sich die Forschung zumeist mit der Adaption bekannter Clustering-Algorithmen und Algorithmen aus den Bereichen Mustererkennung und Data-Mining.

Die Forschung zur Aggregation und Fusion von Kontexten untersucht Verfahren, die zur Verknüpfung abstrakterer Informationen dienen können. Dazu gehören logikbasierte Verkettungsalgorithmen und andere Deduktionsverfahren. Unter der Fusion von Daten versteht man im Allgemeinen die Zusammenfassung von gleichartigen Daten zum Zweck der Genauigkeitsverbesserung oder Fehlerreduktion. Aggregation bezeichnet dagegen die Verknüpfung verschiedenartiger Daten, um daraus neue Erkenntnisse zu gewinnen.

Bei der Kontextaggregation und -fusion werden im Gegensatz zur Kontexterkenntnis nicht Kontexte aus Umweltdaten abgeleitet, sondern verschiedene Kontexte zusammengefasst, um zu umfassenderen Beschreibungen der aktuellen Situation zu gelangen. Zusammenfassend werden die Algorithmen zur Kontexterkenntnis und Kontextaggregation und -fusion, auch kurz Kontextalgorithmen, genannt.

Zentrales Ziel dieser Arbeit ist es Qualitäts sicherungsmechanismen zu etablieren, die es ermöglichen die Erkennungsraten von Kontexten zu verbessern. Dies bedeutet, Qualitätsmaße für Kontexte zur Verfügung zu stellen, die direkt von Kontextalgorithmen ausgenutzt werden können oder Kontexte gezielt auszufiltern, die bei der Anwendung bestimmter Kontextalgorithmen zu Fehlern führen können.

Kontextkommunikation

Im Bereich Kontextkommunikation werden die unterschiedlichen Aspekte der Kommunikation in ubiquitären Informationsumgebungen untersucht, er stellt einen der umfangreichsten Forschungsbereiche im Ubiquitous Computing dar. Von zentralem Interesse sind hier Fragestellungen zu Ad-hoc-Netzwerken wie Routingverfahren und Dienstfindung. Aber auch neue energiesparende Medienzugriffsprotokolle, die speziell den Anforderungen in Ubiquitous Computing Systemen gerecht werden, stehen hier im Fokus. Daneben werden auch neue Protokolle zur Basisbandübertragung untersucht. Damit deckt die Forschung in diesem Bereich die Adaption von Verfahren ab, die den klassischen Schichten 1-3 des ISO/OSI-Basisreferenzmodells [191] entsprechen.

Die Kontextkommunikation wird von der vorliegenden Arbeit in Bereichen berührt, die sich mit der Modellierung der Interaktion von Geräten in ubiquitären Informationsumgebungen beschäftigen, soweit diese bestimmte Kommunikationsparadigmen vorgeben.

Kontextmanagement

Kontextmanagement stellt den jüngsten der hier vorgestellten Forschungsbereiche dar. Die vorliegende Arbeit ist in diesem Bereich angesiedelt. Sie hat zum Ziel, mit der Entwicklung eines Kontextmanagementsystems zur Qualitätssicherung von Kontexten, einen zentralen Beitrag zur aktuellen Forschung zu leisten.

Die Forschung in diesem Bereich erstreckt sich neben den Qualitätssicherungssystemen für Kontexte [28, 24, 187] auf effiziente Verwaltungsstrukturen für Kontexte, wie zum Beispiel zentrale und dezentrale Kontextspeicher (Repositories) [82, 31]. In jüngster Zeit ist hier allerdings die Qualitätssicherung in das zentrale Interesse gerückt.

2.1.3 Anwendungsbereiche dieser Arbeit

Der primäre Anwendungsbereich der Ergebnisse dieser Arbeit sind große ubiquitäre Informationsumgebungen. Ubiquitäre Informationsumgebungen sind räumliche Bereiche, in denen Ubiquitous Computing Systeme arbeiten und entsprechende Dienste bereitstellen. Die Forschung, die dieser Ausarbeitung zu Grunde liegt, adressiert spezielle Probleme, die vornehmlich mit der Skalierung dieser Informationsumgebungen in Zusammenhang stehen und sich auf die Qualität von Kontextdaten auswirken.

Die hier entwickelten Verfahren sollen die möglichen Anwendungsfelder nicht einschränken und sind daher auch nicht auf spezielle Umgebungen zugeschnitten oder setzen diese für eine effiziente Funktion voraus. Allerdings treten einige der adressierten Probleme nur in Umgebungen einer gewissen Größe und Komplexität zutage, so dass manche der entwickelten Verfahren auch nur hier ihren vollen Nutzen entfalten können.

Das zentrale Anwendungsbeispiel dieser Arbeit, das AwareOffice, stammt aus dem Bereich *computerunterstütztes kooperatives Arbeiten* (engl.: Computer Supported Cooperative Work (CSCW)). Die Anwendbarkeit der Ergebnisse beschränkt sich allerdings keinesfalls auf den Einsatz in Büroumgebungen.

Im folgenden Abschnitt 2.1.4 werden die speziellen Eigenschaften der Anwendungsdomäne allgemeiner großer ubiquitärer Informationsumgebungen diskutiert und erste Implikationen daraus identifiziert.

2.1.4 Eigenschaften der Anwendungsdomäne

Die Anwendungsdomäne, in der das hier vorgestellte Qualitätssicherungssystem für Kontexte zum Einsatz kommen soll, zeichnet sich durch unterschiedliche typische Charakteristika aus, die den Entwurf des vorliegenden Systems beeinflusst haben.

Im Gegensatz zu heute in der Forschung gängigen Versuchsaufbauten beinhalten produktiv eingesetzte ubiquitäre Informationsumgebungen eine große Anzahl von Geräten. Wissenschaftliche Versuche in der Kontextverarbeitung werden heute üblicherweise in Testumgebungen durchgeführt, in denen maximal einige Dutzend kontextsensitive Geräte zur Verfügung stehen. Dagegen werden sich typische ubiquitäre Informationsumgebungen auf ganze Gebäude, öffentliche Plätze oder sogar noch größere Gebiete ausdehnen können, die dann erheblich mehr einzelne Geräte umfassen werden. Zählungen, die in der Vorbereitung zu dieser Arbeit durchgeführt wurden, haben ergeben, dass sich im Durchschnitt im Bereich eines Büroarbeitsplatzes mehr als 130 Gegenstände befinden, die in Zukunft potentiell computerisiert sein könnten. Hierzu gehören Büromöbel wie Schränke, Schreibtische und Rollcontainer, Arbeitsmaterialien wie Stifte, Boardmarker, Computer, Telefone, Lampen und Unterlagen in Form von Ordnern, Büchern und Ablagen, sowie persönliche Gegenstände, die an den Arbeitsplatz mitgebracht werden. Die Ergebnisse dieser Studie finden sich im Anhang B dieser Arbeit. Somit können sich bereits in einem einzelnen Büro mit zwei bis drei Arbeitsplätzen mehrere hundert computerisierte Gegenstände befinden, die eine ubiquitäre Informationsumgebung bilden.

Der Großteil der Geräte, die eine ubiquitäre Informationsumgebung bilden ist mobil. Diese können von den Benutzern in den Bereich einer aktiven Umgebung hinein bewegt oder aus diesem entfernt werden. Damit ist im Gegensatz zu überwachten Forschungsumgebungen die Art und Anzahl der Geräte, die eine ubiquitäre Informationsumgebung bilden weder statisch noch genau vorhersagbar.

Es ist davon auszugehen, dass Ubiquitous Computing Systeme von einer Vielzahl verschiedener Hersteller bereitgestellt werden. Die Anwendungen und die verwendete Hardware werden daher nicht vollständig aufeinander abgestimmt sein. Als Folge ergibt sich die Notwendigkeit definierter Schnittstellen für die Kommunikation und Interaktion dieser Geräte, die dann einen Standard für ubiquitäre Informationsumgebungen bilden.

Sind Standards etabliert und können damit beliebige Geräte in einer ubiquitären Informationsumgebung interagieren, ergeben sich dadurch komplexe Verarbeitungszusammenhänge. Die Interaktion von Geräten geschieht über die Kommunikation von Kontexten, die wiederum von den Geräten weiterverarbeitet und zusammengefasst werden. Dadurch sind die Verarbeitungszusammenhänge und Abhängigkeiten einzelner Kontexte ohne zusätzliche Mechanismen nur noch schwer oder gar nicht nachvollziehbar.

Geräte in ubiquitären Informationsumgebungen haben beschränkte Ressourcen. Durch ihre Einbettung in zumeist mobile Gegenstände sind die Geräte auf eine batteriebasierte Energieversorgung angewiesen oder müssen die Energie, die sie benötigen, der Umwelt direkt, zum Beispiel durch Solarzellen, entnehmen. Da die Computer in Ubiquitous Computing Systemen möglichst für den Benutzer unsichtbar bleiben sollen, ist eine der wichtigsten Anforderungen an eine Batterieversorgung, dass diese weitgehend wartungsfrei ist. Das bedeutet, dass der Benutzer möglichst selten oder besser sogar niemals während der Lebenszeit eines Gerätes die Batterien auswechseln muss. Dies bedingt, dass alle Komponenten eines Ubiquitous Computing Gerätes so energieeffizient wie möglich ausgelegt und genutzt werden müssen. Als Folge der Energiebeschränkung ergibt sich damit auch gleichzeitig eine Beschränkung der Rechenleistung, der Speicherkapazität und der verwendbaren Kommunikationsbandbreite. Bei der Entwicklung von Algorithmen und Managementverfahren für Ubiquitous Computing Anwendungen müssen diese Restriktionen beachtet werden.

Skalierung	Große Anzahl Geräte in einer Umgebung
Dynamik	Geräte betreten und verlassen ständig die Umgebung
Heterogenität	Nicht einheitliche Hersteller von Hard- und Software
Komplexität	Nicht planbare Abhängigkeiten bei der Kontextverarbeitung
Ressourcen	Beschränkte Rechenleistung, Speicherplatz, Kommunikationsbandbreite und Energie

Tabelle 2.1. Übersicht der Eigenschaften der Anwendungsdomäne des Kontextmanagementsystems

Tabelle 2.1 fasst die Eigenschaften der Anwendungsdomäne des hier entwickelten Kontextmanagementsystems noch einmal zusammen.

2.2 Kontextbegriffe

Eine Schwierigkeit, der sich die Forschung auf dem Gebiet der ubiquitären Informationsumgebungen gegenüber sieht, ist das Fehlen einer allgemein gültigen Definition des Kontextbegriffes. Dieser Abschnitt gibt einen Überblick über die Entwicklung des Kontextbegriffes innerhalb des Ubiquitous Computing und die eng mit diesem verbundene Architektur von kontextsensitiven Systemen. Neben der Frage, was unter Kontext zu verstehen ist und wie dieser in ubiquitären Informationsumgebungen benutzt wird, wird die Frage behandelt, welche Anforderungen zukünftige kontextsensitive Systeme erfüllen müssen.

2.2.1 Kontext

Die Bezeichnung „Kontext“ (lat. contextus: verflochten, fortlaufend) ist der Linguistik entlehnt. Sie beschreibt dort den sinngebenden Zusammenhang in dem ein Wort oder Satzfragment gebraucht wird. In Bußmanns „Lexikon der Sprachwissenschaft“ [25] findet sich die folgende Definition von Kontext:

Als umfassender Begriff der Kommunikationstheorie bezeichnet K [ontext] alle Elemente einer Kommunikationssituation, die systematisch das Verständnis einer Äußerung bestimmen: den verbalen und non-verbalen (z.B. mimischen) K., den aktuellen K. der Sprechsituation und den sozialen K. der Beziehung zwischen Sprecher und Hörer, ihrem Wissen und ihren Einstellungen.

Im Forschungsfeld des Ubiquitous Computing erscheint der Begriff der *Kontextsensitivität* (engl. context-awareness) zum ersten Mal 1994 in einem Artikel von Bill Schilit und Marvin Theimer [143]. Die erste Definition folgt wenig später in einer eigenständigen Veröffentlichung mit dem Titel „Context-aware computing applications“ [141]. Seit dieser Zeit sind zahlreiche Versuche unternommen worden, sich dem Kontextbegriff zu nähern. Die ersten Definitionen charakterisieren sich durch ihren Versuch Kontext durch Enumeration seiner Komponenten zu beschreiben: Schilit und Theimer verstehen unter Kontext die Lokation und die Identität von Objekten und Personen und die Änderungen, die diese über die Zeit erfahren [143]. Die Identität von Personen in der Umgebung des Benutzers, die Tageszeit und die Jahreszeit werden von Brown, Bovey und Chen [23] als Kontext gesehen.

Andere Definitionen lösen sich von der reinen Beschreibung durch physikalische Größen wie Ort, Zeit, räumliche Nähe (Co-Lokation), Temperatur usw. und bedienen sich einer abstrakteren Sicht von Kontext. Diese Definitionen beinhalten zumeist die *Situation* als Komponente des Kontextes, ohne diesen neuen Begriff genauer zu definieren. In diese Gruppe fallen zum Beispiel die Definitionen von Ward et al. [174], Franklin und Flaschbart [58] und Dey [50]. Dabei benutzen sowohl Ward et al., als auch Franklin und Flaschbart den Begriff *Situation* quasi synonym für Kontext. Im Gegensatz dazu definiert Dey [50]:

Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves.

und differenziert damit zum ersten Mal die umgebende Situation, die beteiligten Instanzen (Personen, Objekte, Orte) und Interaktionen. Diese Definition ist vielleicht die bekannteste und weit verbreitetste Definition des Kontextbegriffes im Bereich Ubiquitous Computing. Dey veröffentlichte sie im Rahmen seiner Dissertation, in der er eines der ersten Toolkits für die Erstellung und den Betrieb von kontextsensitiven Anwendungen entwickelte. Sie bildete einige Zeit den Ausgangspunkt für die Forschung auf dem Gebiet der Kontexterfassung und Kontextverarbeitung.

Auffallend bei allen hier vorgestellten Definitionen ist, dass sie Kontext aus einer Sicht beschreiben, die auf der menschlichen Kognition beruht. Eine Unterscheidung zwischen dem vom Menschen wahrgenommenen *kognitiven Kontext* und den Kontextinformationen, die einem informationstechnischen System zur Verfügung stehen, findet an dieser Stelle nicht statt. Als Folge ist in der Literatur oft nicht eindeutig ersichtlich, ob die Autoren von Kontext im kognitiven Sinne oder von in verarbeitbaren Daten darstellbaren Kontexten sprechen. Einigkeit besteht weitgehend darüber, dass eine vollständige Abbildung des kognitiven Kontextes in informatisch verarbeitbaren Kontextinformationen nicht möglich sein wird. Speziell nicht messbare Faktoren, die zum kognitiven Kontext beitragen, wie zum Beispiel die Intention eines Benutzer bei der Benutzung eines Dienstes, tragen hierzu bei. Die hieraus entstehenden Probleme werden durch Kontextmodelle adressiert.

2.2.2 Kontextsensitivität

Die Grundidee des Ubiquitous Computing – das in den Hintergrund Treten der Technologie während dem Benutzer unaufdringlich ein Mehrwert erbracht wird – ist eng mit der Verarbeitung von Kontexten verbunden. Um ubiquitäre Informationsumgebungen Realität werden zu lassen, müssen sich die darin agierenden Geräte für den Benutzer möglichst unauffällig verhalten, aber dennoch zusätzliche Dienste bereitstellen, wenn diese benötigt werden. Dies wird im Allgemeinen als kontextsensitives Verhalten bezeichnet.

Für Kontextsensitivität finden sich in der Literatur eine Vielzahl sich ähnelnder Definitionen [141, 23, 126, 135, 50]. Beispielhaft soll auch hier die Definition von Dey [50] stehen:

A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.

Diese Definition kontextsensitiver Systeme, fasst frühere Ansätze zusammen, bei denen zum Teil noch Unterscheidungen zwischen Systemen, die Kontextinformationen benutzen und solchen, die durch Adaption auf Kontexte reagieren, gemacht wurden.

Dey geht in seiner Definition stark auf den Aspekt der Relevanz der Information für den Benutzer ein. Da aus heutiger Sicht nicht alle kontextsensitiven Systeme zwingend eine Benutzerschnittstelle haben müssen und die Dienste und Informationen in ubiquitären Informationsumgebungen dem Benutzer nur in den seltensten Fällen von einer einzelnen Anwendung bereitgestellt werden, wollen wir für die vorliegende Arbeit einen allgemeineren Ansatz wählen:

Definition 2.1 (Kontextsensitivität) *Ein System oder eine Anwendung ist kontextsensitiv, wenn es Kontextinformationen verarbeitet und sein inneres und/oder äußeres Verhalten aufgrund der verarbeiteten Kontextinformation ändern kann.*

Die Relevanz für den Benutzer, welche Dey in seine Definition einfließen lässt, ist von großer Wichtigkeit, um den Benutzer eines Systems nicht mit Informationen zu überfluten [7, 86]. Sie kann jedoch als Eigenschaft des Kontextes verstanden werden und hat damit auf die Kontextsensitivität einer verarbeitenden Anwendung keinen Einfluss. Deys Definition beinhaltet damit bereits eine Unterscheidung verschiedener Kontextklassen, die eigentlich einer konkreten Modellierung von Kontexten innerhalb des verarbeitenden Systems zuzurechnen ist.

2.2.3 Architektur

In neueren Arbeiten zum Thema Kontextverarbeitung in ubiquitären Informationsumgebungen ist zu erkennen, dass sich zunehmend eine differenziertere Sicht auf Kontext durchsetzt. Ein Blick auf die Architektur der meisten bekannten kontextsensitiven Anwendungen verdeutlicht jedoch die Quelle der unterschiedlichen Auffassungen und Definitionen von Kontext und Kontextsensitivität die heute in der Forschung gebraucht werden. Wie in Abbildung 2.1 zu sehen ist, bestehen kontextsensitive Anwendungen, wie wir sie heute kennen, zumeist lediglich aus drei eindeutig differenzierbaren Blöcken, die auch in der Forschung und Entwicklung separat adressiert werden. Diese sind die Kommunikation oder das Netzwerk, die Kontextverarbeitung, die heute auch oft Kontext-Middleware genannt wird, und die Anwendung.

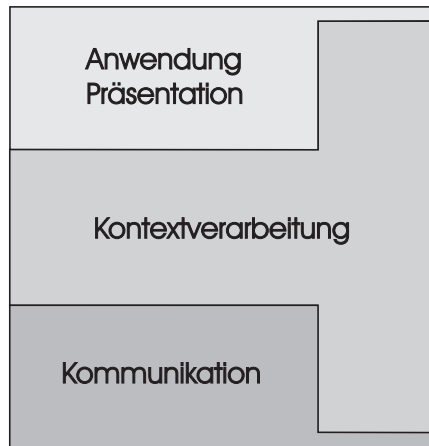


Abbildung 2.1. Einfache Architektur heutiger kontextsensitiver Anwendungen

Die Architektur dieser Anwendungen und damit auch der Systeme, die aus diesen Anwendungen bestehen, ist stark monolithisch. Blöcke, die unterschiedliche Funktionen übernehmen, lassen sich identifizieren, allerdings oft nicht eindeutig trennen. Meist fällt auf, dass das (semantische) Kontextmodell, welches Teil des Kontextverarbeitungsblockes ist, den zentralen Kern des Systems darstellt. Es beeinflusst sowohl die Ausgestaltung der Kommunikation zwischen diesen An-

wendungen, als auch die Funktionalität und Benutzerschnittstellen der Anwendungen, die auf ihm beruhen. Eine klare Trennung von funktionaler Verarbeitung von Kontexten in Form von Daten, die zum Beispiel über ein Netzwerk kommuniziert werden, von der Interpretation von Kontextdaten zur Situationserkennung und Ableitung abstrakterer Kontexte und von der Aufbereitung der kontextuellen Information und ihrer Präsentation gegenüber dem Benutzer, findet meist nicht statt.

Hier zeigt sich die Schwierigkeit im Umgang mit dem Kontextbegriff in der Gestaltung und Umsetzung von realen Anwendungen und Systemen. Die gängigen Definitionen von Kontext differenzieren nicht zwischen dem gesamten facettenreichen kognitiven Kontext in dem eine Anwendung arbeitet und der Information, die der Anwendung als Kontextinformation zur Verfügung steht.

Dabei lassen sich deutliche Unterschiede hinsichtlich der Kontextdefinitionen feststellen, die auf die unterschiedlichen Forschungsschwerpunkte der verschiedenen Arbeiten zurückzuführen sind. Teilbereiche, in denen sich unterschiedliche Kontextbegriffe etabliert haben, sind:

- Kontexte in der Mensch-Maschine-Interaktion (adaptive Benutzerschnittstellen)
- Darstellung von semantischen Kontextinformationen und deren Verarbeitung (semantische Kontextmodelle)
- Algorithmische Kontextgewinnung und Kontextverarbeitung (Fusion und Aggregation)
- Systematische Kontextverarbeitung (Kodierung, Kommunikation, Kontextmanagement, Qualitätssicherung)

Kontexte in der Mensch-Maschine-Interaktion werden hauptsächlich dazu verwendet, Benutzerschnittstellen adaptiv zu gestalten. Definitionen aus diesem Bereich abstrahieren nur wenig vom kognitiven Kontext, da diese als Entscheidungsgrundlage für die Adaption der Schnittstellen dienen soll.

Die semantische Kontextmodellierung fokussiert die Interpretation und Verknüpfung von Kontexten. Bei den hier verwendeten Definitionen steht allerdings auch die inhaltliche Bedeutung der Kontexte im Vordergrund.

Kontextdefinitionen, die in der algorithmischen Kontextverarbeitung gebräuchlich sind, adressieren sowohl funktionale, als auch inhaltliche Aspekte von Kontext. Diese Definitionen dienen dazu, den Übergang zwischen Sensordaten und Kontexten durch die algorithmische Verarbeitung, meist Klassifizierung, abzudecken.

Für die systematische Kontextverarbeitung stehen funktionale Aspekte von Kontext im Zentrum der Definitionen. Auch hier findet allerdings durch die begrifflich unscharfe Trennung oft eine Vermischung mit inhaltlichen Aspekten statt.

Zusammenfassend lässt sich sagen, dass die heutigen einfachen Architekturen von kontextsensitiven Systemen nicht mehr ausreichen, um allen Aspekten von Kontext, die für die Entwicklung eines effizienten Gesamtsystems notwendig sind, zu berücksichtigen.

2.3 Kontextmodellierung

Dieser Abschnitt gibt einen Überblick über die in der Forschung relevanten Kontextmodellierungen und Arbeiten in diesem Bereich. Die Kontextmodellierung bildet aufgrund der im vorigen Abschnitt beschriebenen Architektur in heutigen Systemen oft den Rahmen, der Kontextkommunikation, Kontextverarbeitung und Kontextmanagement bestimmt. Daher wird nun zunächst auf diese Modelle eingegangen, bevor in den folgenden Abschnitten die konkreten Verfahren, die bei der Kommunikation, der Verarbeitung und dem Management von Kontexten zum Einsatz kommen, betrachtet werden.

In der Linguistik werden meist drei Dimensionen des Kontext unterschieden: der extra-linguistische oder situative Kontext, das Hintergrundwissen der Teilnehmer und der unmittelbare Kontext, die linguistische Umgebung der Äußerung [149]. Ähnlich unterscheiden Forscher im Bereich Ubiquitous Computing verschiedene Arten von Informationen, die zum Kontext beitragen: Informationen, die Eigenschaften oder Konfigurationen der Umgebung repräsentieren, werden oft als Meta-Information bezeichnet [63, 96, 185]. Auch hier ist die Benennung nicht immer eindeutig. Davon unterschieden werden Informationen, die aus der Umgebung sensorisch oder durch die Eingabe eines Benutzers gewonnen werden können [144, 30]. Die Übergänge sind dabei oft fließend, was wiederum zu Unklarheiten in der Benennung zwischen verschiedenen Forschungsarbeiten führt.

Allgemein lassen sich zwei Klassen von Kontextmodellen unterscheiden: semantische Kontextmodelle, in deren Mittelpunkt die Umsetzung der Bedeutungs-zuordnung von Kontexten steht und technische oder funktionale Kontextmodelle, die sich mit der Modellierung der Kontextverarbeitung auf informationstechnischer Ebene beschäftigen.

Wie in den vorigen Abschnitten bereits angeklungen ist, stellen zur Zeit die semantischen Kontextmodelle den Hauptteil der Modellierungen von Kontext in Ubicomp. Funktionale Kontextmodelle sind oft Bestandteil ganzer Systemarchitekturen und als solche mit den in diesen Systemen eingesetzten semantischen Modellen eng verknüpft.

2.3.1 Semantische Kontextmodelle

Semantische Kontextmodelle ordnen den erkannten Kontexten einer kontextsensitiven Anwendung eine semantische Bedeutung zu. Sie bilden die Basis für die Interpretation von Kontexten und deren Verknüpfung.

In der Literatur finden sich verschiedene Modellierungsansätze für semantische Kontexte, die hier kurz eingeführt werden sollen. Dabei werden die zentralen Eigenschaften der Modelle anhand von konkreten Vertretern dieser Ansätze erläutert. Eine Übersicht, die auch weitere Modelle berücksichtigt, findet sich zum Beispiel bei Strang und Linnhoff-Popien [155].

Die Diversität der hier verfolgten Ansätze ist der Grund dafür, dass im Rahmen der vorliegenden Arbeit ein Kontextmanagementsystem vorgeschlagen wird, das von der semantischen Modellierung der Kontexte vollkommen unabhängig ist. Auf diese Weise kann erreicht werden, dass in unterschiedlichen ubiquitären Informationsumgebungen die optimalen semantischen Kontextmodelle gewählt werden können, ohne dass tiefgreifende Anpassungen im Kontextmanagementsystem nötig werden.

Unter den hier vorgestellten Modellierungsansätzen haben sich in den letzten Jahren mehr und mehr die objekt- und ontologiebasierten Modelle etabliert. Sie stellen heute den Hauptteil, der in der Literatur beschrieben und in ubiquitären Informationsumgebungen eingesetzten semantischen Kontextmodelle dar.

Logikbasierte Modelle

Zu den frühesten wissenschaftlich beschriebenen Kontextmodellen gehören die logikbasierten Modelle. Bereits 1993 veröffentlichte McCarthy seine „Notes on Formalizing Context“ [115]. Logische Modelle wie dieses, nähern sich der Bedeutung von Kontexten durch die Anwendung von logischen Verknüpfungen, die den Aufbau von Deduktionsketten erlauben. Schlussfolgerungen, die auf diesen Verknüpfungen beruhen, können dann automatisiert gezogen werden. Dabei wird dann die Bedeutung eines Kontextes um neue Fassetten angereichert. Ein einfaches Beispiel für diese Eigenschaft logischer Kontextmodelle ist die Transitivität der Implikation: *Wenn $A \rightarrow B \wedge B \rightarrow C$ dann gilt auch $A \rightarrow C$.*

Bezogen auf die Verarbeitung von Kontexten führt dies zu Modellen, die sich verschiedene Verfahren der Logik, wie zum Beispiel Hornklauseln, zu Nutze machen, um Zusammenhänge zwischen Kontexten zu modellieren [158]. Ebenfalls ein logikbasiertes Kontextmodell wurde von Gray und Salbe vorgestellt [63]. In diesem Modell wird Prädikatenlogik erster Ordnung eingesetzt, um die Beziehungen und Aussagen von Kontexten zu formulieren.

Schlüssel-Wert Modelle

Schlüssel-Wert Modelle stellen Kontexte der Anwendung als ein Paar, bestehend aus einem Bezeichner, dem Schlüssel, und einem Wert zur Verfügung. Der Bezeichner determiniert dabei die Bedeutung eines Kontextes. Der Vorteil dieser Modelle ist, dass sie extrem einfach zu implementieren und zu verwalten sind. Dagegen steht allerdings der Nachteil, dass auf Listen von Schlüssel-Wert Paaren keine effizienten Algorithmen zur Kontextverknüpfung implementiert werden können. Daher eignen sie sich vornehmlich für abgeschlossene Umgebungen in denen nur eine geringe festgelegte Anzahl von Kontexten verarbeitet wird. Ein prominentes Beispiel für ein Schlüssel-Wert Modell findet sich im bekannten ParcTab-Projekt des Xerox Parc [142]. Es wurde 1994 unter dem Titel „Context-Aware Computing Applications“ [141] veröffentlicht.

Profilbasierte Modelle

Profilbasierte Ansätze bei der Modellierung von Kontexten nutzen die Vorteile von Markup-Sprachen wie der *Standard Generic Markup Language (SGML)* und

daraus abgeleiteten Sprachen wie *XML*. Für diese Sprachen existieren leistungsstarke Parser und Interpreter.

Beispiele für profilbasierte Kontextmodelle wurden unter anderem von Indulska et al. [85] oder Held et al. [69] vorgeschlagen. Das Modell von Indulska et al. basiert auf dem CC/PP Standard [169]. Sowohl die Bedeutung der Kontexte als auch die Beziehungen von Kontexten werden hier als Profile abgelegt. Das Modell von Held et al., CSCP (Comprehensive Structured Context Profiles) basiert auf dem Ressource Description Framework (RDF). Es fokussiert stärker auf die Strukturierung von Kontexten und die Abbildung von Benutzerpräferenzen direkt in der Kontextmodellierung.

Nachteil profilbasierter Kontextmodelle ist ihre schlechte dynamische Anpassbarkeit, da die Profile, welche die Kontextmodellierung enthalten, als statische Dokumente vorliegen, die für eine Anpassung bearbeitet oder ersetzt werden müssen.

Objektorientierte Modelle

In objektorientierten Kontextmodellen werden Kontexte als Datenobjekte mit Eigenschaften und eigenen Methoden modelliert, über welche auf die Felder der Kontextobjekte zugegriffen werden kann. Die Objektklasse determiniert dabei die semantische Bedeutung des Kontextes. Beziehungen zwischen Kontexten können durch Klassenvererbung realisiert werden, wie sie aus objektorientierten Programmiersprachen bekannt ist. Komplexere Beziehungen von Kontexten werden durch Verzeigerung der Objekte in den Eigenschaftsfeldern realisiert. Die in Forschungsarbeiten beschriebenen Systeme implementieren diesen generellen Aufbau in unterschiedlicher Tiefe. Nicht alle Komponenten sind in jedem Modell vorhanden.

Die bekanntesten Vertreter dieser Klasse von Kontextmodellen sind sicherlich das „Context Toolkit“ von Anind Dey [50, 48, 49] und „TEA“ von Albrecht Schmidt et al. [146, 148]. Neuere Arbeiten auf diesem Gebiet umfassen „Context Fabric“ von Jason Hong und James Landay [82, 81, 83], ein Agenten System, in dem Kontextobjekte verwaltet werden, die „Reconfigurable Context-Sensitive Middleware (RCSM)“ von Stephan Yau et al. [179, 181, 180] und das „Java Context Awareness Framework (JCAF)“ von Jacob Bardram [9], eine JAVA API, die Kontextobjekte implementiert.

Ontologiebasierte Modelle

Modelle dieser Klasse verwenden Ontologien zur Speicherung semantischer Zusammenhänge. Ontologien [162, 161] sind Strukturen, die so genannte Konzepte der realen Welt abbilden und diese in Form von Beschreibungen und Beziehungen speichern können. Diese Konzepte können vereinfacht als überlagerte Inklusionsrelationen zwischen Beschreibungen erklärt werden. Die Bedeutung eines einzelnen Elementes einer Ontologie ergibt sich durch seine Einbettung in die ontologische Struktur.

Ontologien sind für die Wissensrepräsentation vorteilhaft, da sie auf einfache Weise, durch das Hinzufügen von Beschreibungen und Konzepten, erweitert werden können. So kann auch die Granularität der Kontextauflösung leicht verfeinert werden. Nachteilig am Einsatz von Ontologien ist ihre Größe und schlechte Komprimierbarkeit.

Konzeptuelle Modelle wurden zum Beispiel von J. Coutaz et al. [45] und M. Khedr und A. Karmouch [96] vorgestellt. Daneben existieren in der Literatur vollständige Architekturen und Systeme, die auf ontologischer Kontextverarbeitung basieren. Dazu gehört „PACE“ von K. Hinricksen et al. [73, 78, 75], „GAIA“ von der Gruppe um Roy Campbell [138, 131] und „SOCAM“ von Tao Gu et al. [66, 65, 172, 67], die sich als Kontext-Middleware bezeichnen. Andere bekannte ontologiebasierte Kontextmodelle sind Teil der „Context Broker Architecture (CoBrA)“ von Harry Chen et al. [32, 31] und des „Gadgetware Architectural Style (GAS)“ von Achilles Kameas et al. [94, 39, 40, 41]. Diese Systeme verstehen sich, wie die Namen bereits andeuten, mehr als globale Architekturen zur semantischen Kontextverarbeitung.

In neuester Zeit zeichnet sich in der Forschung, die sich mit der ontologischen Modellierung von Kontexten in ubiquitären Informationsumgebungen beschäftigt, ein Trend hin zur Integration der Ontologien des *Semantic Web* [21, 150] ab.

2.3.2 Funktionale Kontextmodelle

Funktionale Kontextmodelle beschäftigen sich mit der Modellierung der Kontextverarbeitung auf informationstechnischer Ebene. Diese Modelle sind, wie einige der bereits vorgestellten semantischen Kontextmodelle oft Teil ganzer Systeme

oder Architekturen für ubiquitäre Informationsumgebungen. Funktionale Modelle unterscheiden sich hauptsächlich dadurch von semantischen Kontextmodellen, dass bei ihnen nicht die Bedeutung der Kontexte im Vordergrund steht, sondern deren Darstellung innerhalb der informatischen Systeme, aus denen eine ubiquitäre Informationsumgebung aufgebaut ist.

Zwei Typen von funktionalen Kontextmodellen können unterschieden werden: Repräsentationsmodelle, die vorwiegend die effiziente Darstellung der Kontexte für die Kommunikation und Verarbeitung in den Geräten einer ubiquitären Informationsumgebung adressieren und Managementmodelle, welche die Verwaltungsstrukturen für Kontextdaten in den verteilten Systemen ubiquitärer Informationsumgebungen fokussieren.

Repräsentationsmodelle

Repräsentationsmodelle für Kontext beschäftigen sich mit der effizienten Darstellung von Kontexten. Sie werden, durch die monolithische Architektur der meisten heutigen kontextverarbeitenden Systeme, von den Anforderungen der verwendeten semantischen Modelle und denen der ihnen zu Grunde liegenden Kommunikationssysteme geprägt.

Neben den obigen Kontextsystemen finden sich Beispiele für Repräsentationsmodelle in den Arbeiten zum „FuzzySpace“ von Albrecht Schmidt [147, 144] und zu „ConCom“ von Albert Krohn et al. [106]. Beide Systeme favorisieren eine tupelbasierte Repräsentation von Kontexten.

Die objektorientierten Systeme bedienen sich zur Repräsentation meist der Mechanismen, die durch die eingesetzten Programmiersprachen wie Java oder C++ vorgegeben werden, um Kontextobjekte zu kodieren. In den ontologiebasierten Systemen werden zur Darstellung der Kontexte, wie auch in den profilbasierten und Schlüssel-Wert Systemen, oft sehr einfache Ansätze gewählt. Zum Beispiel werden die Kontexte als Listen gehalten, die einen Identifikator und die Kontextdaten als Zeichenkette oder numerischen Wert enthalten [141].

Managementmodelle

Kontextmanagementmodelle modellieren die Verwaltung von Kontextdaten in Ubiquitous Computing Systemen. Mit der fortschreitenden Entwicklung im Be-

reich der Kontextverarbeitung gewinnt das effiziente Management von Kontexten und deren Verarbeitung und Kommunikation mehr und mehr an Bedeutung.

Auch die bekannten Managementmodelle sind meist in die, bereits im Abschnitt 2.3.1 über semantische Kontextmodelle, vorgestellten Systeme eingebettet. Darüber hinausgehende Arbeiten in diesem Bereich beschäftigen sich vorwiegend mit Kontextmanagement unter dem Gesichtspunkt der Kontextqualität [74, 95].

Da das Management von Kontexten und deren Qualitätssicherung das zentrale Thema der vorliegenden Arbeit ist, wird auf die relevanten Arbeiten in diesem Bereich, die über die reine Modellierung hinausgehen, in Abschnitt 2.6 gesondert eingegangen.

2.4 Kontextkommunikation

Die Kommunikation von Kontexten zwischen Ubiquitous Computing Geräten bildet die Grundlage für die Entstehung von ubiquitären Informationsumgebungen. Sie ist daher für alle Betrachtungen, die Vorgänge in diesen Umgebungen betreffen, von höchster Relevanz.

Aufgrund des Umfangs dieses Forschungsbereiches innerhalb des Ubiquitous Computing ist es an dieser Stelle unmöglich, einen vollständigen Überblick über den Stand der Forschung zu geben. Daher beschränkt sich dieser Abschnitt darauf, für die vorliegende Arbeit relevante Aspekte der Kontextkommunikation einzuführen. Eine eingehende Analyse möglicher Kommunikationskonzepte und deren Eignung für den Einsatz in ubiquitären Informationsumgebungen findet sich in Abschnitt 3.2 dieser Arbeit, daher werden diese Aspekte hier ausgeklammert.

2.4.1 Mobile Ad-hoc-Netzwerke

Der Austausch von Kontexten zwischen den mobilen Geräten einer ubiquitären Informationsumgebung geschieht zumeist per Funk. Für die Kommunikation innerhalb der Infrastruktur dieser Umgebungen wird oft auf klassische kabelgebundene Netzwerklösungen auf der Basis von TCP/IP zurückgegriffen, da diese Netze in den meisten Fällen bereits vorhanden sind und ohne zusätzliche Kosten für Installation und Betrieb für die Kommunikation von Kontextdaten mit genutzt werden können.

Durch die hohe Dynamik in ubiquitären Informationsumgebungen, welche durch die Mobilität der Teilnehmer bedingt ist, sind für die Realisierung der Funkkommunikation zwischen den mobilen Knoten Ad-hoc-Netzwerke von zentralem Interesse [113, 160]. Für den Aufbau und die Funktion von ubiquitären Informationsumgebungen sind dabei besonders Routing- und Zugriffsverfahren von Bedeutung, da diese die möglichen konzeptuellen Kommunikationsmodelle determinieren, die sich in realen Systemen umsetzen lassen.

Studien zu Routingprotokollen für mobile Ad-hoc-Netzwerke finden sich zum Beispiel bei Hong et al. und Qin und Kunz [84, 128]. Ein aktueller Vergleich der Zugriffsprotokolle für mobile Ad-hoc-Netzwerke wurde Mitte 2006 von Kumar et al. veröffentlicht [107].

Daneben existieren in der Literatur konzeptuelle Vorschläge für den Aufbau von Kontextkommunikationssystemen, die zum Teil von den verwendeten Routing- und Zugriffsprotokollen unabhängig sind [138, 17, 49].

2.4.2 Relevante Aspekte für diese Arbeit

Durch Kommunikation werden Kontexte in ubiquitären Informationsumgebungen verteilt und den verschiedenen Teilnehmern zugänglich gemacht. Für die Bewertung der Kontextqualität sind Einflüsse durch die Kommunikation interessant, da die räumlichen und zeitlichen Abhängigkeiten der Daten durch deren Kommunikation aufgelöst werden können.

Für die vorliegende Arbeit sind zum einen Aspekte der Kommunikationsmodellierung von Bedeutung, diese werden in Abschnitt 3.2 analysiert. Zum anderen kann die Erkennungsrate kontextsensitiver Anwendungen durch die zeitlichen und räumlichen Einflüsse der Kommunikation beeinflusst werden. Dieser Aspekt wird in Abschnitt 3.8 eingehend behandelt.

2.5 Kontextverarbeitung

Die Kontextverarbeitung stellt die zentrale Komponente kontextsensitiver Systeme dar. In diesem Bereich werden Algorithmen und Verfahren untersucht, welche die Erkennung von Kontexten auf der Basis von sensorisch erfassten Daten oder anderer Kontexte ermöglichen.

2.5.1 Klassifikation und Aggregation

Die Kontextgewinnung ist dabei fast immer ein Klassifikationsprozess durch den ein Satz Eingangsdaten einem Kontext, der durch das Kontextmodell mit einer bestimmten Bedeutung verknüpft ist, zugeordnet wird. Dabei kommen verschiedenste Klassifikationsalgorithmen zum Einsatz, deren Auswahl zumeist von den Anforderungen an die Kontexterkenkung, oder der Art der zu verarbeitenden Sensordaten abhängt.

Beispiele für häufig verwendete Algorithmen zur Kontextgewinnung aus Sensordaten sind: C4.5 für die Erkennung von Aktivitäten im Büro oder Haushalt [22] oder zur allgemeinen Aktivitätserkennung [8], Bayes'sche Netze [102], Hidden Markov Models (HMM) [42], neuronale Netze [127], unscharfe Logik [108] und selbstorganisierende Karten (SOM) und K-Means [110].

Für die Ableitung aus bereits bestehenden Kontexten, der Kontextaggregation, werden Verfahren eingesetzt, welche die Verknüpfung klassifizierter Daten unterstützen. Zu nennen sind hier unter anderem regelbasierte Verfahren [57], unscharfe Logik [27] und ontologiebasierte Verfahren [41].

2.5.2 Fusion

Bei der Fusion werden vornehmlich gleichartige Daten verarbeitet. Sie dient dazu, die Genauigkeit und Zuverlässigkeit zu erhöhen. Sowohl Kontexte als auch die Sensordaten aus denen sie abgeleitet werden, können fusioniert werden wo dies sinnvoll erscheint.

Gängige Verfahren für die Fusion unkorrelierter Daten sind in diesem Gebiet zum Beispiel logisches UND, ODER, K aus N, Mehrheitsentscheid und Neyman-Pearson [165, 139, 166, 134, 152]. Daneben existieren weitere Algorithmen, die speziell für die Fusion von korrelierten Daten entwickelt wurden [93, 33]. Einen vollständigeren Überblick über bekannte Fusionsalgorithmen enthält zum Beispiel „Adaptive fusion of correlated local decisions“ von Chen und Ansari [33].

2.5.3 Relevante Aspekte für diese Arbeit

Die algorithmische Verarbeitung von Kontexten und Sensordaten bildet den zentralen Kern kontextsensitiver Systeme. Soll die Qualität von Kontexten beurteilt

werden, ist die Kontexterkennungsrate der Anwendungen das ausschlaggebende globale Maß, da die Fähigkeit der Anwendungen zur korrekten Adaption oder Reaktion von der korrekten Erkennung der Umgebungskontexte abhängen.

Da Kontext eine Abstraktion der realen Situation darstellt und somit für Anwendungen, die auf der Basis von Kontexten arbeiten, die ursprünglichen Ausgangsinformationen nicht mehr in voller Tiefe zur Verfügung stehen, ist die Beurteilung der Leistung der eingesetzten Erkennungsalgorithmen ein wichtiger Baustein bei der Bestimmung der Qualität eines Kontextes. Dieser Aspekt wird in Abschnitt 3.8 der vorliegenden Arbeit ausführlich analysiert.

2.6 Kontextmanagement

Kontextmanagementsysteme sind erst in den letzten Jahren als eigener Bereich in den Fokus der Forschung gerückt. Zuvor erreichten kontextverarbeitende Systeme nicht die Größe und Komplexität ab der dedizierte Managementstrukturen für Kontexte von Nöten sind.

Kontextmanagementsysteme definieren die Verwaltungsstrukturen für Kontexte in ubiquitären Informationsumgebungen. Das zentrale Interesse gilt dabei der Sicherung und Verbesserung der Qualität der Kontexterkennung.

Vorschläge für Kontextmanagementsysteme wurden in den letzten Jahren unter anderem von Mendes et al., Henricksen et al. und Glassey et al. gemacht. Mendes et al. [117] konzentrieren sich dabei auf die Vorverarbeitung von Kontexten und verstehen darunter in erster Linie die Koordination zwischen Kommunikation und der Kontextverarbeitung. Der Ansatz von Henricksen et al. in „Generating Context Management Infrastructure from Context Models“ [76] zielt bereits direkt auf die Abbildung von Kontexteigenschaften in den vorgeschlagenen Managementstrukturen, um so die Gesamtqualität im System zu verbessern. Auf ähnliche Weise adressiert „Towards a Middleware for Generalised Context Management“ von Glassey et al. [61] die Abbildung der Eigenschaften von Kontexten, um diese für die Aggregation von Daten nutzbar zu machen.

2.6.1 Qualität der Kontexterkennung

Die Kontexterkennung ist in Ubiquitous Computing Systemen die Grundlage für das dynamisch-adaptive Verhalten der Anwendungen. Sie ist die Voraussetzung

für die Umsetzung der Ideen von Marc Weiser [176] und Donald Norman [123]. Aus Sicht der Benutzer, die sich in ubiquitären Informationsumgebungen bewegen und mit diesen interagieren, sollen die eingesetzten Technologien möglichst unauffällig und vor allem auch unaufdringlich ihre Dienste bereitstellen. Dabei ist die Dienstgüte das ausschlaggebende Moment.

Unterlaufen bei der Evaluierung des Umgebungskontextes Fehler, die zu unangebrachten Aktionen oder Reaktionen der kontextsensitiven Anwendungen führen, sind diese, da sie nicht der tatsächlichen Situation angemessen sind, für den Benutzer auffällig. Die Anwendung kann so nicht in den kognitiven Hintergrund treten. Passieren Fehler zu häufig, wird außerdem die Akzeptanz der Benutzer gegenüber den unzuverlässigen Diensten sinken. Die Dienstgüte kontextsensitiver Anwendungen ist also direkt von der Qualität der Kontexterkennung abhängig.

Wie bereits in Abschnitt 2.5 dargestellt wurde, handelt es sich bei der algorithmischen Kontexterkennung im Wesentlichen um eine Klassifikation von Eingangsdaten, durch die Umgebungssituationen beschrieben werden. Dabei ist klar, dass die entstehenden Kontexte die reale Welt nicht in ihrer vollen Informationstiefe wiedergeben können. Dies beruht zum einen auf der inhärenten Ungenauigkeit jeder verwendeten Sensorik und zum anderen auf der Abstraktion, die der Kontextgenerierungsprozess darstellt.

Ungenauigkeiten bei der sensorischen Umwelterfassung können, genau wie falsche Eingangskontexte, bei der Verarbeitung zur Fehlklassifikation führen und so falsche oder unangebrachte Reaktionen der Anwendung nach sich ziehen. Die Ungenauigkeit der Kontextaussage aufgrund der Abstraktion kann ebenfalls dazu führen, dass der erkannte Kontext nicht mit der realen Situation übereinstimmt. Hinzu kommen Einflüsse, die nicht von sensorischen Systemen erfasst werden können, wie zum Beispiel die Intension des Benutzers einer ubiquitären Informationsumgebung [64].

Diese Problematik ist in der Forschung zu kontextverarbeitenden Systemen seit langem bekannt. Bereits die ersten Arbeiten in diesem Bereich von John McCarty [115] und Bill Schilit et al. [142, 143] erwähnen die Relevanz von Kontextzuverlässigkeit und Kontextungenauigkeiten am Rande.

Spätere Kontextmodelle gehen fast ohne Ausnahme auf einzelne für ihre konkrete Umsetzung relevante Parameter ein, welche die Kontextqualität beeinflussen können. Dabei beschränken sie sich allerdings häufig auf die Adressierung von konkreten Problemstellungen, die sich aus den vorgeschlagenen Anwendungsbeispielen ergeben.

Darüber hinaus existieren Modelle, die sich in erster Linie mit der Handhabung nicht perfekter Kontexte auseinander setzen, wie die Arbeiten von A. Dey et al. [51, 53, 52], K. Henricksen und J. Indulska [74, 71] und T. Strang et al. [157, 156].

Dey et al. schlagen vor, den Benutzer bei der Beurteilung der Kontexterken-
nung aktiv einzubeziehen. Diese ist zum einen problematisch, da dies dem Prin-
zip der Unsichtbarkeit und Unaufdringlichkeit zuwiderläuft, zum anderen ist eine
aktive Mediation von Kontexten in großen ubiquitären Informationsumgebungen
kaum praktikabel, wenn mehrere hundert kontextsensitive Anwendungen Benut-
zereingaben verlangen. Henricksen und Indulska schlagen vor, Parameter welche
die Unzulänglichkeiten der Kontextinformation beschreiben, in ihr ontologisches
Kontextmodell zu integrieren und so diese Unzulänglichkeiten auf semantischer
Ebene abzufangen. Problematisch dabei ist, dass eine semantische Interpretati-
on jedes Kontextes notwendig ist, bevor die Managementinformationen für eine
Anwendung verfügbar sind. Ein systemnahes effizientes Kontextmanagement für
große Umgebungen lässt sich durch den so vervielfachten Aufwand bei der Kon-
texthandhabung kaum realisieren. Auf ähnliche Weise nähern sich Strange et al.
dem Problem. Hier wird allerdings sogar eine parallele Management-Ontologie
zu der vorgeschlagenen semantischen Ontologie eingeführt, deren Zusammen-
führung den Gesamtaufwand weiter erhöht.

Jenseits dieser Ansätze, die bestehende semantische Modelle erweitern, exis-
tieren auch einige Arbeiten, die sich konkret mit der Umsetzung und Handhabung
von Kontextqualität befassen. Im Mittelpunkt dieser Arbeiten steht die Definiti-
on von Kontextqualität und der allgemeinen Parameter, welche diese beschrei-
ben. Im Gegensatz zu den modellintegrierten Ansätzen ist die Einführung des
Begriffes der Kontextqualität als selbständiger Instanz der erste Schritt auf dem
Weg zu einer Trennung von inhaltlicher Modellierung und Qualitätsmodellierung
von Kontexten. Hauptunterscheidungsmerkmal zu den vorigen Ansätzen ist, dass
Qualitätsparameter modelliert werden, welche die Qualität aller im System vor-
handener Kontexte gleichermaßen erfassen können.

Ebling et al. sprechen in „Issues for Context Services for Pervasive Compu-
ting“ [55] von *Quality of Information (QoI)* und führen erste Metriken ein, welche
Kontextqualität vergleichbar machen sollen. Buchholz et al. [24] verwenden den
Begriff *Quality of Context (QoC)* und stellen eine allgemeine Definition für das
vor, was sie unter Kontextqualität verstehen:

Quality of Context (QoC) is any information that describes the qua-
lity of information that is used as context information. Thus, QoC

refers to information and not to the process nor the hardware component that possibly provide the information.

Diese Definition bildet eine sehr gute Grundlage, um sich dem Thema zu nähern, da sie auf die Prozess- und Hardwareunabhängigkeit des Qualitätsmaßes hinweist. Leider verfehlen die von Buchholz et al. definierten Qualitätsparameter dieses Ziel, da diese mit der „Korrektheitswahrscheinlichkeit (Probability of Correctness)“ und der „Vertrauenswürdigkeit (Trust-worthiness)“ zwei auf Wahrscheinlichkeiten beruhende Maße definieren, die per se nur Prozesse und nicht einzelne Kontexte beschreiben können.

Dem „Metadaten“-Ansatz von Hönle et al. liegt ein ähnliches Verständnis von Kontextqualität zu Grunde. Allerdings wird hier zwischen abstrakten Eigenschaften der Kontexte als Qualitätsparameter unterschieden; diese sind *Metadaten zu Datenanbietern*, *Metadaten zu Datenobjekten* und *Metadaten zu Attributwerten*. Eine weitere Konkretisierung dieser Klassen von Metadaten findet nur anhand von Beispielen statt.

Razzaque et al. betrachten Kontextqualität unter dem Begriff *Quality of Contextual Information (QoCI)* [133]. Ihre Arbeit fokussiert auf die Bereitstellung von Qualitätsparametern, welche die Kategorisierung, also die semantische Zuordnung von Kontexten erleichtern. Sie stellen die Trennung der Qualitätsmodellierung und der semantischen Modellierung in den Vordergrund, allerdings werden hier keine konkreten Qualitätsparameter definiert, die zur Erreichung der formulierten Ziele dienen sollen.

Krause und Hochstätter [104] bauen auf der Arbeit von Buchholz et al. [24] auf und wandeln deren Definition von Kontextqualität leicht ab:

Quality of Context (QoC) is any inherent information that describes context information and can be used to determine the worth of the information for a specific application. This includes information about the provisioning process the information has undergone („history“, „age“), but not estimations about future provisioning steps it might run through.

Sie verzichten in ihrer Definition auf die explizite Prozessunabhängigkeit des Qualitätsmaßes. Dafür führen sie Beispiele für mögliche Qualitätsparameter an

und beziehen sich damit auf Einflüsse des Verarbeitungsprozesses. Insgesamt erscheint diese Definition damit sogar schwächer als die ursprüngliche von Buchholz et al.

Keine der vorgestellten Arbeiten konkretisiert Kontextqualität zu einem anwendbaren Konzept. Eine algorithmische Annäherung an das Thema „Qualitätsmaße für Kontext“ fehlt ebenso wie eine funktionale Definition von Qualitätsparametern und ihrer Anwendung. Keine dieser Arbeiten kann daher den Nutzen der Verarbeitung der Kontextqualität anhand konkreter Ergebnisse belegen.

Die theoretischen Vorarbeiten werden in dieser Arbeit daher genutzt, um zu dieser Konkretisierung zu gelangen und ein System zur Qualitätssicherung von Kontexten umzusetzen. Dazu wird nun zunächst der Begriff Kontextqualität für die vorliegende Arbeit in Anlehnung an Buchholz et al. definiert:

Definition 2.2 (Kontextqualität) *Kontextqualität sind Informationen, auf welchen Metriken definiert werden können, die eine quantitative Beurteilung der Qualität von Kontexten bezüglich bestimmter Qualitätsparameter zulassen. Kontextqualität bezieht sich auf einzelne Kontexte, nicht auf die Prozesse, durch die Kontexte bereitgestellt werden. Kontextqualität ist ein globales Maß, das auf alle Kontexte gleichermaßen anwendbar ist; ihrer Bemessung werden für alle Kontexte die gleichen Qualitätsparameter zu Grunde gelegt.*

Diese Definition beinhaltet sowohl die Unabhängigkeit der Metriken zur Bestimmung der Kontextqualität von der inhaltlichen Bedeutung der beurteilten Kontexte, als auch die Individualität des Qualitätsmaßes bezüglich jedes einzelnen Kontextes durch den Ausschluss von prozessbeschreibenden Maßen. Dadurch kann die so definierte Kontextqualität auf jeden Kontext einer ubiquitären Informationsumgebung angewendet werden und beschreibt, im Gegensatz zu statistischen Maßen, Eigenschaften dieses speziellen Kontextes.

2.7 Zusammenfassung

Der zentrale Forschungsbereich dieser Arbeit ist das Kontextmanagement in ubiquitären Informationsumgebungen. In diesem Kapitel wurde zunächst in den allgemeineren Forschungsbereich des Ubiquitous Computing eingeführt und das Themengebiet der vorliegenden Arbeit gegen die angrenzenden Themen Kontextmodellierung, Kontextverarbeitung und Kontextkommunikation abgegrenzt und

die grundlegenden Eigenschaften der Anwendungsdomäne des Kontextmanagementsystems, welches in dieser Arbeit entwickelt wird, analysiert.

Diese sind:

- Skalierung: Eine große Anzahl Geräte sind in einer Umgebung vorhanden.
- Dynamik: Geräte betreten und verlassen ständig die Umgebung.
- Heterogenität: Geräte stammen von unterschiedlichen Herstellern.
- Komplexität: Nicht planbare Abhängigkeiten in der Verarbeitung von Kontexten.
- Ressourcen: Stark beschränkte Ressourcen der Teilnehmer einer Umgebung.

Weiterhin wurde der, für die Forschung im Bereich ubiquitärer Informationstechnologien, zentrale Kontextbegriff erörtert und die, für die vorliegende Arbeit gültige Definition der Kontextsensitivität von Anwendungen, eingeführt. Die Betrachtung der Architektur heute gängiger Ubiquitous Computing Systeme zeigt einen monolithischen Aufbau, der von den semantischen Kontextmodellen dominiert wird.

Im zweiten Teil des Kapitels werden die für diese Arbeit relevanten Themenbereiche Kontextmodellierung, Kontextkommunikation und Kontextverarbeitung anhand wichtiger Forschungsarbeiten vorgestellt und vertieft. Abschließend wurde der Stand der Forschung im zentralen Bereich dieser Arbeit – Kontextmanagement – erörtert und die dort relevanten Arbeiten analysiert. Auf dieser Basis wurde der wichtige Begriff der Kontextqualität für diese Arbeit definiert. Diese Definition beinhaltet auch erste Hinweise auf die Anforderungen, die ein Kontextmanagementsystem zur Sicherung der Kontextqualität in ubiquitären Informationsumgebungen erfüllen muss.

3 Modell

Dieses Kapitel stellt das Modell einer ubiquitären Informationsumgebung vor, in der attributierte Kontexte kommuniziert und verarbeitet werden. Modelliert werden dabei funktionale Aspekte der Kontextgenerierung, Kontextverarbeitung und Kommunikation von Kontexten. Das hier vorgestellte Modell beruht auf der mehrjährigen Arbeit mit und an ubiquitären Informationsumgebungen und spiegelt die dabei gesammelten Erfahrungen wieder.

Wie in Kapitel 2 gezeigt wurde, existiert eine Vielzahl von Modellen die ubiquitäre Informationsumgebungen beschreiben. Der Großteil dieser Modelle fokussiert die semantische Darstellung und Verarbeitung von Kontexten. Aufgrund der Architektur dieser Systeme als monolithischer Block, der Kommunikation, Kontextmanagement und auch Benutzerschnittstellen einschließt, berücksichtigen diese Modelle die unterschiedlichen Aspekte einer ubiquitären Informationsumgebung mit unterschiedlicher Gewichtung.

Da sich die vorliegende Arbeit ausdrücklich auf den Bereich des Kontextmanagements konzentriert, werden in der hier vorgestellten Modellierung nur Aspekte berücksichtigt, die für das Management von Kontextinformationen relevant sind. Von allen Aspekten, welche die semantische Kontextverarbeitung betreffen, wird bewusst abstrahiert, um eine möglichst breite Kompatibilität des hier vorgestellten Kontextmanagementsystems mit verschiedenen semantischen Kontextverarbeitungsmodellen zu gewährleisten.

Zuerst werden die Thesen dieser Arbeit in Beziehung zu der folgenden Modellierung gesetzt. Dann wird das, dieser Arbeit zu Grunde liegende Modell der Kommunikation und Verarbeitung von Kontexten vorgestellt. Dabei wird das zentrale Konzept der *Kontextattribute* als Basis für das Kontextmanagement eingeführt. Der zweite Teil dieses Kapitels beschäftigt sich mit der Modellierung dieser Kontextattribute und ihrer Funktion.

3.1 Thesen und Modelle

Die Hauptthese dieser Arbeit besagt, dass die Qualität der Kontexterkennung und damit die Qualität der Dienste, die in einer ubiquitären Informationsumgebung erbracht werden, durch geeignete Managementmechanismen verbessert werden kann. Voraussetzung für effiziente Qualitätssicherungsmechanismen ist eine systematische und formale Ausgestaltung der Repräsentation, Verarbeitung und Kommunikation von Kontexten.

Die Modellierung geht zunächst auf die *Subthese 2* dieser Arbeit ein, indem ausgehend von einem Kostenmodell für die Kommunikation und Verarbeitung von Kontexten, die grundlegenden Anforderungen an eine formal und systematisch ausgestaltete ubiquitäre Informationsumgebung erarbeitet werden. Daraufhin wird die Kontextrepräsentation, -verarbeitung und -kommunikation entsprechend der Anforderungen formalisiert.

Dabei wird die vormalig oft monolithische Architektur bekannter Kontextmodelle zu Gunsten eines Schichtenmodells erweitert, in dem zwischen generellen und allgemeinen Eigenschaften von Kontexten auf Systemebene, den Kontextattributen, und der inhaltlichen Bedeutung von Kontexten, ihrer semantischen Komponente, unterschieden wird.

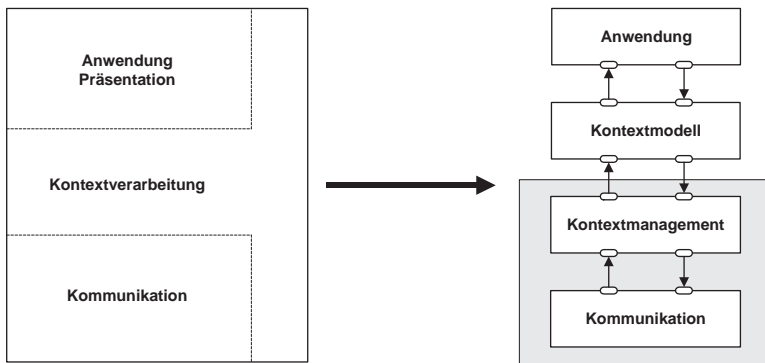


Abbildung 3.1. Übergang vom monolithischen zum Schichtenmodell

Abbildung 3.1 zeigt den Übergang zwischen diesen Modellen. In der linken Hälfte der Abbildung ist eine typische monolithische Architektur zu sehen, wie

sie heute Grundlage vieler Ubiquitous Computing Systeme ist. Rechts ist das in dieser Arbeit favorisierte Schichtenmodell dargestellt. Der grau hinterlegte Teil des Schichtenmodells enthält die systemnahen Schichten, in denen Kontextverarbeitung in einem informatischen Sinne stattfindet. Die darüber liegenden Schichten verarbeiten Kontexte basierend auf deren semantischer Bedeutung. Die zentrale *Subthese 1* dieser Arbeit wird durch die Einführung einer separaten Kontextmanagementschicht adressiert.

Dabei repräsentieren Kontextattribute in den unteren, systemnahen Schichten, Eigenschaften von Kontexten, deren Information von einem Kontextmanagementsystem genutzt werden kann. Das Kontextmanagement stellt Verfahren bereit, welche die Kontexterkennungsrate auf der semantischen Ebene verbessern. Dazu wird anhand der Attributwerte die Qualität eines Kontextes bestimmt. Die Schicht der semantischen Kontextverarbeitung kann hierbei Vorgaben zu der minimalen, von der Anwendung akzeptierbaren Kontextqualität machen. Diese werden im Kontextmanagement umgesetzt, so dass Kontexte inakzeptabler Qualität abgewiesen werden. Die semantische Kontextverarbeitung einer Anwendung erhält somit nur qualitativ hochwertige Kontexte, wobei die Anwendung die Qualitätsmaßstäbe individuell ihren Erfordernissen anpassen kann.

Die Umsetzung eines Qualitätsmanagementsystems für Kontexte erfordert die Einführung von Maßen für die Kontextqualität, daher ist es notwendig die hierfür relevanten Komponenten einer ubiquitären Informationsumgebung formal zu modellieren und so die Konsistenz der eingeführten Maße sicherzustellen. Diese Komponenten sind im Einzelnen der Kontext selbst, als zentrale Datenstruktur des Informationsaustausches innerhalb der Umgebung, und seine Attribute, als Basisdaten für das Kontextmanagement. Daneben ist der Austausch und die Verarbeitung dieser Kontexte in der ubiquitären Informationsumgebung zu modellieren, da hierdurch grundlegende Prozesse spezifiziert werden, welche die Qualität der Daten mit beeinflussen. Das Kostenmodell aus [189] dient hier als Grundlage für die zu treffenden Modellierungsentscheidungen.

3.2 Kostenmodell der Kontextkommunikation und Kontextverarbeitung

Aus dem oben eingeführten systemnahen Kontextmodell sowie bekannten Modellen der Kommunikation in Netzwerken und bereits in der Literatur beschriebenen Kontextmodellen, wurde im Rahmen dieser Arbeit ein Kostenmodell für

die Kommunikation und Kontextverarbeitung in ubiquitären Informationsumgebungen abgeleitet [189]. Dieses Kostenmodell hatte zum Ziel, die verschiedenen möglichen Modellierungsansätze hinsichtlich ihrer Kosten vergleichbar zu machen und so Hinweise für die effiziente Modellierung der Kommunikation und Verarbeitung von Kontexten zu geben.

3.2.1 Untersuchte Kommunikations- und Verarbeitungsmodelle

Basis des Kostenmodelles sind zwei Kostenklassen: Kosten, welche durch die Kommunikation entstehen und Kosten, welche durch die Verarbeitung von Daten entstehen. Untersucht wurden verschiedene alternative Kommunikations- und Verarbeitungsmodelle für ubiquitäre Informationsumgebungen, die zu unterschiedlichen Gesamtkosten für den Betrieb einer ubiquitären Informationsumgebung führen.

Untersucht und hinsichtlich ihrer Kosten wurden verglichen:

Vollständig zentralisierte Systeme , wie zum Beispiel die *Reconfigurable Context-Sensitive Middleware (RCSM)* von Stephan Yau et al. [179], das *Java Context Awareness Framework (JCAF)* von Jakob Bardram [9], *Context Fabric (ConFab)* von Jason Hong [83], das *Pervasive Autonomic Context-aware Environment (PACE)* von Karen Henriksen et al. [73, 75] und die *Context Broker Architecture (CoBrA)* von Harry Chen et al. [31].

Zentralisiert vermittelnde Systeme , wie zum Beispiel das *Context Toolkit* von Anind Dey [50, 49], die *Service-Oriented Context-Aware Middleware (SOCAM)* von Tao Gu et al. [66], *Gaia* von Manuel Roman [138] und das *SO-LAR System*, einer Implementierung des Context Fusion Network (CFN) von Guanling Chen [30].

Dezentrale Systeme , wie zum Beispiel der *Gadgetware Architectural Style (GAS)* von Achilles Kameas et al. [94] und der *FuzzySpace* von Albrecht Schmidt [147].

Für jedes dieser Modelle wurden die zu erwartenden Kosten für Kommunikation und Datenverarbeitung hinsichtlich eines einheitlich definierten Basisprozesses (siehe [189]) berechnet und verglichen. Neben der grundlegenden Systemorganisation wurde bei den Berechnungen unterschieden, ob die Systeme sub-

skriptionsbasiert arbeiten, oder ob sie Kontextdaten anfragebasiert zur Verfügung stellen.

3.2.2 Ergebnisse der Kostenrechnung

Das Ergebnis der Kostenbetrachtung aus [189] ist in Tabelle 3.1 zusammengefasst. Die einzelnen Systemkonfigurationen sind aufsteigend nach den durchschnittlichen Kosten eines Basisprozesses sortiert. $\bar{K}(B)$ bezeichnet die durchschnittlichen Kosten eines Basisprozesses. $\bar{K}'_{n_{seg}}$ bezeichnet die erste Ableitung der Kostenentwicklung unter der Veränderung der Anzahl der Netzwerksegmente in einer ubiquitären Informationsumgebung. Je kleiner dieser Wert ist, desto stabiler sind die Kosten eines Systems gegenüber einer Skalierung der Anzahl der Netzwerksegmente. \bar{K}'_d bezeichnet die erste Ableitung der Kostenentwicklung unter einer Veränderung der Dynamik in einem System. Die Dynamik wird dabei definiert durch die Anzahl von Geräten, die eine Umgebung verlassen oder in diese Eintreten. Auch hier zeigt ein kleinerer Wert an, dass die Kosten eines Systems gegenüber einer Skalierung der Dynamik stabiler sind.

Es zeigt sich, dass offenbar Systeme, die vorwiegend subskriptionsbasiert arbeiten, solchen, die anfragebasiert arbeiten, hinsichtlich der durchschnittlichen Kosten überlegen sind. Als kostengünstigste Modellierung einer ubiquitären Informationsumgebung zeigt sich das dezentrale subskriptionsbasierte System.

System Organisation	$\bar{K}(B)$	$\bar{K}'_{n_{seg}}$	\bar{K}'_d
dezentral subskriptionsbasiert	3,6	7,2	2,0
zentral vermittelt subskriptionsbasiert	4,8	9,6	6,0
zentralisiert subskriptionsbasiert	4,9	9,8	3,0
dezentral anfragebasiert	5,0	10,0	0,0
zentralisiert anfragebasiert	5,6	11,2	2,0
zentral vermittelt anfragebasiert	9,6	19,2	2,0

Tabelle 3.1. Kosten im Vergleich [189]

Neben diesem Kostenvorteil bietet die dezentrale Lösung weitere Vorteile für die Realisierung von ubiquitären Informationsumgebungen. Einer der wichtigsten dieser Vorteile ist, dass es in einem dezentral organisierten System keine zentrale Instanz gibt, deren Ausfallen das Gesamtsystem funktionsunfähig machen kann.

Hinzukommt, dass nur in dezentral organisierten Systemen die Möglichkeit eines echten peer-to-peer Betriebes besteht, für den keinerlei Infrastruktur nötig ist. Das heißt, nur dezentrale Systeme sind in der Lage die Ad-hoc-Ausbildung von ubiquitären Informationsumgebungen zu unterstützen, wo immer Artefakte aufeinander treffen und miteinander kommunizieren können.

Basierend auf den Ergebnissen dieser Kostenanalyse wird im Folgenden das Modell für die Kommunikation und Verarbeitung von Kontexten entworfen, welches als Grundlage für das in dieser Arbeit entwickelte Kontextmanagementsystem dient. Dieses Modell baut auf bestehenden Ansätzen für die Modellierung dezentraler ubiquitärer Informationsumgebungen auf und erweitert diese zu einem funktionalen System, das sich bereits im Einsatz bewährt hat.

3.3 Modell der Kontextkommunikation und Kontextverarbeitung

In diesem Abschnitt wird das Modell der Kommunikation und Verarbeitung von Kontexten eingeführt, welches die Basis für das, in dieser Arbeit entwickelte Kontextmanagementsystem bildet. Es modelliert den Austausch und die Verarbeitung von Kontexten, ohne das semantische Verarbeitungsmodell festzulegen oder einzuschränken. Zuerst wird dabei auf die Charakteristik der Kontextverarbeitung in einer ubiquitären Informationsumgebung eingegangen, dann wird der Kontext-Raum als das Modell der Kommunikation in diesen Umgebungen eingeführt.

Wie im vorigen Abschnitt gezeigt wurde, stellt ein dezentrales Modell, in dem Kontexte überwiegend subskriptionsbasiert ausgetauscht werden, den Idealfall für ein Ubiquitous Computing System dar. Das hier entwickelte Modell setzt diese Erkenntnis konsequent um.

3.3.1 Anforderungen an Kommunikation und Kontextverarbeitung in ubiquitären Informationsumgebungen

In diesem Abschnitt werden die Anforderungen zusammengefasst, welche die Kommunikation und Verarbeitung von Kontextdaten in ubiquitären Informationsumgebungen erfüllen sollen. Dieser Anforderungskatalog bildet die Grundlage für die Modellierung einer allgemeinen ubiquitären Informationsumgebung, welche als Ausgangspunkt für die weiteren Überlegungen dieser Arbeit dient.

Bei ubiquitären Informationsumgebungen handelt es sich um verteilte Systeme, in denen Dienste durch die Interaktion der Anwendungen verschiedener Artefakte komponiert werden. Voraussetzung für große Kommunikationsumgebungen dieser Art ist die Unabhängigkeit der Nutzung und Produktion von Kontexten. Ist die Produktion von Kontexten von deren Nutzung unabhängig, können nutzende und produzierende Anwendungen und damit auch Artefakte von verschiedenen Anbietern bereitgestellt werden. Im Gegensatz zu typischen Beispielumgebungen, die zu Demonstrationszwecken zusammengestellt werden, ist es nicht nötig, Produzenten und Nutzer von Kontexten aufeinander abzustimmen. Vielmehr wird davon ausgegangen, dass durch eine genügend große Anzahl ubiquitär vorhandener Artefakte ubiquitäre Informationsumgebungen ad hoc entstehen. Dabei muss gewährleistet sein, dass die Dienste der Artefakte auch ohne vorherige Konfiguration komponierbar sind und so Anwendungen einen Mehrwert für den Benutzer bereitstellen können. Diese a priori unkoordinierte Umgebungskonfiguration, in der Artefakte und Anwendungen nicht aus einer Hand stammen, wird im Allgemeinen als Heterogenität ubiquitärer Informationsumgebungen bezeichnet.

Wenn viele verschiedene Hersteller und Anbieter Hardware und Software bereitstellen, die nicht aufeinander abgestimmt ist aber dennoch zusammenarbeiten soll, ist es von Vorteil, wenn jedes einzelne Gerät über möglichst wenig Informationen über seine Kommunikationspartner verfügen muss, um Daten mit ihnen auszutauschen.

Eine weitere Eigenschaft großer ubiquitärer Informationsumgebungen ist ihre Dynamik. Dynamik einer Umgebung kommt zustande, da der Großteil der aktiven Artefakte, die eine ubiquitäre Informationsumgebung bilden, mobile Gegenstände sind, die von den Benutzern der Umgebung bewegt und manipuliert werden, oder sogar der direkten Interaktion mit der Umgebung dienen. Diese Artefakte können den Bereich einer ubiquitäre Informationsumgebung jederzeit betreten oder verlassen. Außerdem können diese Artefakte, wie bereits beschrieben, ad hoc ubiquitäre Informationsumgebungen durch ihr Zusammentreffen (auf Kommunikationsebene) definieren.

Damit ist weder die Menge der in einer ubiquitären Informationsumgebung zur Verfügung stehenden Kontexte, noch deren Topologie statisch. Als Folge kann sich die Anwendung eines Artefaktes nicht darauf verlassen, dass bestimmte Kontexte oder Kommunikationspartner zur Verfügung stehen. Sie sollte daher so entworfen sein, dass sie eine breite Auswahl an Kontexten nutzen kann, um ihren Dienst zu erbringen. Das Appliance-Prinzip von Donald Norman [123] legt so-

gar nahe, dass jedes Artefakt seine Aufgabe sogar außerhalb einer ubiquitären Informationsumgebung erfüllen können sollte.

Aus den bisherigen Betrachtungen leiten sich konkrete Anforderungen an ubiquitäre Informationsumgebungen ab:

- In einer ubiquitäre Informationsumgebung soll die Produktion und die Nutzung von Kontexten unabhängig sein. Auf diese Weise können Artefakte und Anwendungen aus verschiedenen Quellen zu einer ubiquitären Informationsumgebung zusammengefasst werden. Dabei ist es nicht notwendig diese Artefakte und Anwendungen a priori aufeinander abzustimmen.
- Dienste sollen komponierbar sein. Durch die dynamische Komposition von Diensten auf Anwenderebene kann das volle Potential, der in einer ubiquitäre Informationsumgebung vorhandenen Ressourcen, genutzt werden. Dienste, die modular komponiert werden, ermöglichen die effiziente Wiederverwendung von Informationen (Kontexten) die in einer ubiquitären Informationsumgebung zur Verfügung stehen. Dabei müssen Kontexte nicht für die Verwendung durch mehrere verschiedene Anwendungen mehrfach erzeugt werden.
- Die Kommunikation zwischen den Komponenten einer ubiquitäre Informationsumgebung soll ad hoc erfolgen können. Damit ist das Vorhalten einer Infrastruktur nicht notwendige Voraussetzung für das Entstehen einer ubiquitären Informationsumgebung. Ubiquitäre Informationsumgebungen können dann auch dynamisch, durch das Zusammentreffen von Artefakten auf Kommunikationsebene, gebildet werden.
- Kommunikation in einer ubiquitären Informationsumgebung soll entkoppelt sein. Das bedeutet, die Kommunikation soll nicht auf der Identität¹ der Kommunikationspartner beruhen. Dadurch stehen Kontextdaten potentiell allen Geräten zur Verfügung, die auf das gewählte Kommunikationssystem zugreifen können. Bessere Wiederverwendbarkeit der Daten und damit eine höhere Modularität sind die Folgen.
- Hinsichtlich der zu erwartenden Kommunikationskosten soll das Kommunikationsmodell, welches einer ubiquitären Informationsumgebung zu

¹Identitätsbezogene Kommunikation meint hier sowohl adressbasierte Kommunikation als auch andere Kommunikationsparadigmen, in denen die Datenquelle den Kreis der möglichen Datensenden einschränken kann.

Grunde liegt, dezentral und subskriptionsbasiert organisiert sein. Wie die Kostenrechnung in Abschnitt 3.2 gezeigt hat, ist ein solcher Ansatz im Hinblick auf die entstehenden Kosten optimal.

Im Folgenden wird das Modell der Kommunikation und Kontextverarbeitung in einer ubiquitären Informationsumgebung vorgestellt, welches für diese Arbeit entwickelt wurde und die Basis des hier vorgestellten Qualitätssicherungssystems für die Kontextverarbeitung bildet. Das Modell berücksichtigt die hier erarbeiteten Anforderungen. Es stellt eine mögliche Modellierung ubiquitärer Informationsumgebungen dar und ist Ausgangspunkt für die simulativen wie auch realen Systeme, mit denen im Rahmen der vorliegenden Arbeit gearbeitet wurde.

3.4 Kontext als Datenstruktur

Für die Verarbeitung von Kontexten in ubiquitären Informationsumgebungen existieren eine Vielzahl von Modellen. Der überwiegende Teil dieser Modelle beschreibt die Behandlung von Kontexten auf semantischer Ebene, ihre Generierung aus Rohdaten und ihre Fusionierung und Aggregation zu neuen Kontexten. Das hier vorgestellte Modell der attributierten Kontextverarbeitung integriert sich in diese allgemein akzeptierten Kontextmodelle und erweitert diese um formale Definitionen und Beschreibungen wichtiger Aspekte der Kommunikation und Datenverarbeitung.

Der wichtigste Unterschied zu bestehenden Modellen ist, dass Kontext hier nicht im Sinne der in Abschnitt 2.2 vorgestellten Definitionen verstanden wird, sondern eine systemnahe Sicht gewählt wurde. Die in Kapitel 2 vorgestellten Kontextdefinitionen sind ausnahmslos beschreibender Natur. Sie führen Beispiele an oder beschreiben, was als Kontext einer Anwendung oder einer ubiquitären Informationsumgebung verstanden werden soll. Dabei vermischen sich inhaltliche Aspekte, die semantische Bedeutung, mit informatischen Aspekten wie dem datentypischen Aufbau oder der Kodierung des Kontextes. In dieser Arbeit steht der effiziente Umgang und die informatische Verarbeitung von Kontexten durch Anwendungen in ubiquitären Informationsumgebungen im Mittelpunkt. Daher werden nun zunächst die benötigten Begriffe eingeführt und unter diesem Gesichtspunkt definiert. Dann wird der dieser Arbeit zu Grunde liegende Aufbau einer allgemeinen ubiquitären Informationsumgebung modelliert und auf die Kommunikation von Teilnehmern innerhalb einer solchen Umgebung eingegangen.

3.4.1 Einordnung und Begriffsbildung

Im Fokus dieser Arbeit steht die Entwicklung eines Kontextmanagementsystems, das, wie Abbildung 3.2 zeigt, als systemnahe Schicht die Schnittstelle zwischen der netzwerktechnischen Kommunikation und dem verwendeten semantischen Kontextmodell bildet.

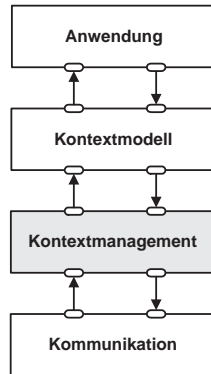


Abbildung 3.2. Kontextmanagement als Schnittstelle zwischen Kommunikation und Kontextverarbeitung

Eine ubiquitäre Informationsumgebung ist ein nicht zwingend zusammenhängender Bereich oder ein Gebiet physikalischer Ausdehnung, in dem Artefakte und optional kontextsensitive Infrastrukturdienste vorhanden sind und miteinander kommunizieren können.

Definition 3.1 (Artefakt) *Ein Artefakt ist ein computerisierter Alltagsgegenstand in einer ubiquitären Informationsumgebung, der eine kontextsensitive Anwendung beinhaltet. Der Begriff wird oft synonym für die Anwendung und ihre Funktionalität verwendet, wenn die Gefahr einer Verwechslung nicht besteht.*

Infrastrukturdienste in einer ubiquitären Informationsumgebung sind kontextsensitive Dienste, die nicht an einen Alltagsgegenstand gebunden sind. Sie kommen nur in speziellen ubiquitären Informationsumgebungen vor, die über eine Infrastruktur verfügen. Im allgemeinen Fall liegt einer ubiquitären Informationsumgebung keine Infrastruktur zu Grunde. Sie kann ad hoc durch das Zusammen-

treffen von Artefakten entstehen, die in der Lage sind, drahtlos direkt (peer-to-peer) zu kommunizieren. Dies bedingt, dass sowohl die Ausdehnung als auch die Lokation einer ubiquitäre Informationsumgebung variabel sein können.

Grundlage für das adaptive Verhalten kontextsensitiver Anwendungen ist die Wahrnehmung der Umwelt durch Sensoren.

Definition 3.2 (Sensordaten) *Sensordaten sind Daten, die ein Artefakt oder ein Infrastrukturdienst durch physikalische Umweltsensoren erfassen kann.*

Zu den Sensordaten können auch Daten gehören, die über eine Benutzerschnittstelle in das System eingegeben wurden. Die Benutzerschnittstelle bildet dabei einen komplexen Sensor, der die Benutzereingabe zurückliefert. Nach der oben gegebenen Definition sind Sensordaten die reinen Daten, die von einem Sensor geliefert werden. Handelt es sich um einen physikalischen Umweltsensor sind dies die Daten, welche an der digitalen Schnittstelle des Sensors zur Verfügung stehen. Im Falle eines analogen Sensors sollen die Daten, die ein angeschlossener Analog-Digital-Wandler (A/D-Wandler) bereitstellt, als Sensordaten verstanden werden.

Sensordaten alleine können praktisch nicht weiter verarbeitet werden, da ihnen wichtige Informationen fehlen, die zu ihrer Interpretation nötig sind. Das führt zum Begriff der Metainformation:

Definition 3.3 (Metainformation) *Metainformation bezeichnet Daten, die nicht innerhalb des Informationssystems selbst gewonnen werden können, sondern stattdessen von außen, zum Beispiel durch den Entwickler, bereitgestellt werden müssen.*

Im Gegensatz zu Sensordaten, die dynamisch aus der Umwelt gewonnen werden, müssen Metainformationen im System bereits bereitstehen. Sie werden zum Beispiel während der Entwicklung einer Anwendung durch den Entwickler implizit oder explizit in der Anwendung verankert. Metainformationen können zur semantischen Interpretation von Sensordaten dienen. Dabei kann es sich zum Beispiel um die Identität eines verwendeten Sensors handeln. So erst ist es möglich einer Messung, also Sensordaten, eine physikalische Einheit zuzuordnen. Andere Beispiele für Metadaten sind der Typ einer Anwendung oder eines Artefaktes oder deren Anwendungsdomäne, falls diese statisch ist.

Die Metainformationen spielen eine wichtige Rolle beim Übergang von physikalischen Größen zu Kontexten. Kontext zeichnet sich dadurch aus, dass er Zusammenhänge der realen Welt beschreibt und diese für Anwendungen nutzbar

macht. Um aus Sensordaten diese Zusammenhänge abzuleiten ist es erforderlich, die Daten der Sensoren mit weiteren Informationen zu kombinieren.

Ein einfaches Beispiel für die Verwendung von Metadaten stammt aus dem AwareOffice. Die Fenster im AwareOffice sind mit Kleinstcomputern ausgestattet die feststellen können, ob das Fenster gerade offen oder geschlossen ist. Diese Information kann das Fenster als Kontext im AwareOffice bereitstellen. Die sensorische Detektion des Zustandes des Fensters geschieht mittels eines einfachen berührungsfreien Kontaktes. Die Sensordaten, welche die Fensteranwendung verarbeiten kann, um daraus einen Kontext zu generieren, beschränken sich also auf den Schaltzustand eines Kontaktes. Der Kontext „das Fenster ist geöffnet“ oder „das Fenster ist geschlossen“, kann nur erzeugt werden, wenn die Information, dass der Sensor an einem Fenster angebracht ist, als Metainformation zur Verfügung gestellt wird. Die Identität eines Artefaktes ist die grundlegendste Metainformation, die nötig ist, um aus Sensordaten Kontexte abzuleiten.

Das hier vorgestellte System wurde für die Verarbeitung von Kontexten auf Systemebene entwickelt. Wie in Abschnitt 2.2 gezeigt wurde, werden in der Forschung sehr unterschiedliche Kontextbegriffe benutzt. Daher ist es nötig exakt zu definieren, was in dieser Arbeit unter Kontext zu verstehen ist.

Die in 2.2 eingeführten Definitionen von Kontext unterscheiden nicht zwischen kognitivem Kontext, semantischem Kontext und Kontextdaten. Daher werden diese Begriffe nun hier für die vorliegende Arbeit definiert, um im Weiteren Missverständnisse auszuschließen.

Definition 3.4 (kognitiver Kontext) *Kontext, wie er vom Menschen wahrgenommen wird. Der kognitive Kontext beinhaltet sowohl messbare physikalische Umweltfaktoren als auch nicht messbare Faktoren, wie die Erfahrungen oder die Intention des Menschen.*

Der kognitive Kontext kann als der tatsächliche, reale Kontext einer Umgebung verstanden werden. Da er ein Phänomen der menschlichen Wahrnehmung ist, kann er nicht direkt in abstrakten Daten, die einem Rechnersystem zugeführt werden können, abgebildet werden.

Ziel der Kontextmodellierung im Ubiquitous Computing ist es, diesen kognitiven Kontext möglichst exakt in einem informationstechnischen System zu repräsentieren und die enthaltenen Informationen für kontextsensitive Systeme zugänglich zu machen. Dies führt zur Definition des semantischen Kontextes:

Definition 3.5 (semantischer Kontext) *Die Repräsentation des kognitiven Kontextes in einem kontextsensitiven System heißt semantischer Kontext. Der semantische Kontext dient der Erfassung der Bedeutung einer Kontextinformation. Semantische Kontextmodelle modellieren diese Bedeutung und die Beziehung zwischen semantischen Kontexten.*

Da der semantische Kontext die Bedeutung und Beziehung von Kontextinformationen beschreibt ist es nötig, diesen von einem technischen Kontextberiff abzugrenzen. Kontexte im technischen Sinne sind in erster Linie Daten, die bestimmte Eigenschaften haben. Eine dieser Eigenschaften ist, dass Teile dieser Daten als semantische Kontexte interpretiert werden können. Aus technischer Sicht steht hier aber weniger die semantische Bedeutung im Vordergrund, sondern vielmehr die Kodierung der Daten und der Aufbau der verwendeten Datentypen. Daraus ergibt sich die Kapselung semantischer Kontexte auf technischer Ebene.

Definition 3.6 (Kontextdaten) *Kontextdaten setzen sich aus mindestens einem Sensordatum und einer Metainformation zusammen. Kontextdaten haben eine semantische Bedeutung bezüglich der aktuellen Situation.*

Kontextdaten sind Daten, die aus Sensordaten und Metainformationen erzeugt werden können. Wie oben beschrieben, unterscheiden sich Kontextdaten von Sensordaten dadurch, dass ihnen durch die Verbindung mit Metainformationen bereits eine Bedeutung zugeschrieben werden kann. Die obige Definition stellt die minimalen Anforderungen an Kontextdaten dar. Kontextdaten können auch in mehrstufigen Prozessen aus einer Vielzahl von verschiedenen Sensordaten und Metainformationen zusammengefügt werden. Somit können sie selbst wieder Kontextdaten enthalten, die der obigen Definition genügen.

Die Kontextdaten können als die Nutzdaten der semantischen Kontextverarbeitung betrachtet werden. Ihre Kodierung, ihr Aufbau und ihre Bedeutung werden in dem verwendeten semantischen Kontextmodell festgelegt. Eines der zentralen Ziele dieser Arbeit ist es, die Unabhängigkeit von semantischem Kontextmodell und Kontextmanagement sicherzustellen. Um Kontextmanagement unabhängig von einem bestimmten semantischen Kontextmodell halten zu können, müssen die Kontextdaten für das Kontextmanagement transparent sein. Dies wiederum bedeutet, dass eine Kontextmanagementschicht nicht auf die semantische Bedeutung eines Kontextes zugreifen kann. Die Funktionalität der Kontextmanagementschicht muss sich also auf allgemeine Eigenschaften der Kontextdaten beziehen, die von der semantischen Bedeutung dieser Daten unabhängig sind.

Definition 3.7 (Kontextattribut) *Ein Kontextattribut ist eine Eigenschaft eines Kontextdatums, die von dessen semantischer Bedeutung unabhängig ist.*

Kontextattribute sind die Basis für das in dieser Arbeit entwickelte Kontextmanagementsystem. Die Unabhängigkeit von Kontextattributen von der semantischen Bedeutung der Kontextdaten soll hier an einem einfachen Beispiel erläutert werden. Im Sinne der obigen Definition kann zum Beispiel das Alter eines Kontextdatums als Kontextattribut aufgefasst werden. Das Alter erfüllt die Anforderungen der Definition, da jedes Kontextdatum ein Alter besitzt und die Bedeutung der Kontextdaten dieses Alter nicht beeinflusst. Offensichtlich ist die Unabhängigkeit von Kontextattributen und Kontextdaten keine symmetrische Beziehung, da die semantische Interpretation der Kontextdaten von ihrem Alter abhängen können und dies im Allgemeinen auch tun werden.

Die vorliegende Arbeit befasst sich mit Kontexten vornehmlich aus einer technischen Sicht. Daher wird der Begriff Kontext hier unter diesem Aspekt definiert:

Definition 3.8 (Kontext) *Kontext bezeichnet die Einheit aus Kontextdaten und Kontextattributen. Des Weiteren besitzt Kontext einen Typ, der durch die enthaltenen Kontextdaten festgelegt wird.*

Damit ist der Begriff Kontext für diese Arbeit eindeutig definiert. Im Gegensatz zu früheren Arbeiten, in denen der Terminus „Kontext“ kognitive Kontexte, semantische Kontexte sowie Kontexte auf informatischer Ebene beschreibt, wird hier eine Vermischung dieser grundlegend unterschiedlichen Kontextbegriffe vermieden. Diese Trennung ermöglicht die systematische Modellierung eines modularen Kontextmanagementsystems, das über geeignete Schnittstellen mit einer Vielzahl verschiedener semantischer Kontextsysteme zusammenarbeiten kann.

Im folgenden Abschnitt wird der detaillierte Aufbau eines Kontextes formal beschrieben.

3.4.2 Aufbau eines Kontextes

Basierend auf der obigen Definition des Kontextbegriffes wird nun der formale Aufbau eines Kontextes innerhalb des hier entwickelten Modells einer ubiquitären Informationsumgebung eingeführt. Kontext wird hier als Datenstruktur modelliert, die innerhalb einer ubiquitären Informationsumgebung erzeugt und

verarbeitet wird. Sie bildet die Grundlage der Interaktion kontextsensitiver Anwendungen und damit die Basis für die Entstehung ubiquitärer Informationsumgebungen.

Sei T eine abzählbare Menge von Kontexttypen, die durch ein semantisches Kontextmodell zur Verfügung gestellt wird. Stellt das verwendete semantische Kontextmodell keine konkreten Typen zur Verfügung, sei T eine abzählbare oder überabzählbare Menge von abstrakten Schlüsseln, die das Wiederauffinden einer Bedeutungszuordnung innerhalb der semantischen Modellierung erlauben. Sei weiter A eine abzählbare Menge von Kontextattributen und D eine abzählbare oder überabzählbare Menge von Kontextdaten.

$$\begin{aligned} T &:= \{\text{Kontexttypen}\} \\ A &:= \{\text{Kontextattribute}\} \\ D &:= \{\text{Kontextdaten}\} \end{aligned}$$

Dann sei ein Kontext C definiert als ein 3-Tupel

$$C := \{t, a, d\} \quad \text{mit} \quad t \in T, \quad a \in A, \quad d \in D. \quad (3.1)$$

Der Kontexttyp dient innerhalb einer ubiquitären Informationsumgebung zur Auswahl von Kontexten. Er verbindet die semantische Modellierung der Bedeutung eines Kontextes mit den entsprechenden Datenstrukturen, welche diese auf der technischen Verarbeitungs- und Kommunikationsebene repräsentieren. Der Typ bildet dabei einen Schlüssel, der eine Klasse von Kontexten identifiziert.

Definition 3.9 (Kontextklasse) *Eine Kontextklasse wird durch den Typ des Kontextes induziert. Die Kontextklasse hat alle Eigenschaften und Attribute eines Kontextes, ohne dass diesen konkrete Werte zugewiesen sind.*

Eine Kontextklasse fasst alle Kontexte zusammen, denen innerhalb eines semantischen Kontextmodells die gleiche Bedeutung zukommt. Ist einem Kontext neben des Typs eine konkrete Belegung seiner Attribute und Kontextdaten zugeordnet, spricht man von einer Kontextinstanz.

Definition 3.10 (Kontextinstanz) *Eine Kontextinstanz ist Kontext einer bestimmten Kontextklasse, dessen Attributen konkrete Werte zugeordnet sind und dessen Kontextdaten einen konkreten Wert besitzen.*

Der Kontext *C* stellt die elementare Datenstruktur des hier eingeführten Kontextmodells dar, auf ihr sind drei Grundoperationen definiert:

- Die erzeugende Operation *C_produce*: *C_produce* instantiiert eine Kontextklasse und belegt die neu erzeugte Kontextinstanz mit konkreten Werten für den Typ, die Kontextattribute und die Kontextdaten. Im Allgemeinen ist *C_produce* eine Operation des semantischen Kontextmodells, die aufgerufen wird, wenn ein neuer Kontext erzeugt wird. Verfügt das semantische Kontextmodell nicht über die Fähigkeit die Kontextattribute selbst zu setzen, können diese an einem Dienstzugangspunkt (engl. Service Access Point, SAP) zwischen semantischem Kontextmodell und Kontextmanagement von der Managementschicht gesetzt werden.
- Die lesende Operation *C_read*: *C_read* liest die Daten aus den Feldern *t*, *a* und *d* einer Kontextinstanz. *C_read* ist eine Operation der Kommunikationsschicht, welche die Kontexte entgegennimmt. Nach dem Auslesen steht der Inhalt der Felder *t*, *a* und *d* an dem SAP der Kontextmanagementschicht zur Verfügung.
- Die löschende Operation *C_delete*: *C_delete* löscht und dereferenziert eine Kontextinstanz vollständig. Die Löschoperation ist eine Operation der Kommunikationsschicht. Sie kann nicht expliziert von den aktiven Komponenten einer ubiquitären Informationsumgebung ausgeführt werden, sondern wird implizit aufgerufen, wenn eine Kontextinstanz nicht mehr über die vorhandenen Kommunikationsmedien zur Verfügung steht.

Durch die Einschränkung der löschenden Operation wird auf Modellebene verhindert, dass Artefakte oder Infrastrukturobjekte einer ubiquitären Informationsumgebung Informationen aktiv zerstören können.

Die Modellierung von Kontext als Datenstruktur innerhalb eines verteilten Kommunikationssystems ist einer der zentralen Beiträge dieser Arbeit. Kontext wird hier soweit konkretisiert, dass er für datenverarbeitende Systeme handhabbar wird. Dies wiederum ist eine zwingende Voraussetzung für die Entwicklung realitätsrelevanter Qualitätssicherungsmechanismen.

3.5 Produzent-Konsument-Schema

Die Entkoppelung von Bereitstellung und Benutzung der Kontextdaten wird im Rahmen des hier vorgestellten Modells mittels eines Produzent-Konsument-Schemas realisiert.

Artefakte in ubiquitären Informationsumgebungen interagieren, in dem sie Kontexte kommunizieren, diese austauschen. In einem dezentral organisierten System verarbeiten die Artefakte diese Kontexte lokal weiter. Dabei kann ein Artefakt entweder Daten bereitstellen oder die Daten gebrauchen, die von einem anderen Artefakt bereitgestellt wurden. Wenn ein Kontext bereitgestellt wird, wird er zu eine unabhängigen Instanz seiner Kontextklasse.

Definition 3.11 (Kontextproduzent) *Ein Artefakt, welches einen Kontext erzeugt und diesen in einer ubiquitären Informationsumgebung zur Verfügung stellt, heißt Produzent dieses Kontextes.*

Wenn Kontextdaten verarbeitet werden, muss dabei die Instanz des Kontextes nicht zwingend zerstört oder verbraucht werden. Mehrere Artefakte können ein und dieselbe Instanz eines Kontextes parallel verarbeiten. In Anlehnung an den Begriff „Kontextproduzent“ heißt ein Artefakt, welches einen Kontext in einer ubiquitären Informationsumgebung aufnimmt und verarbeitet „Kontextkonsument“.

Definition 3.12 (Kontextkonsument) *Ein Artefakt, welches Kontexte aus einer ubiquitären Informationsumgebung aufnimmt und verarbeitet heißt, Konsument dieser Kontexte.*

Ein Artefakt kann zugleich Konsument und Produzent von Kontexten sein. Zum Beispiel trifft dies auf Artefakte zu, die einen Kontext aus anderen Kontexten ableiten. Diese Artefakte sind für die Quellkontexte Kontextkonsumenten, während sie gleichzeitig Produzenten des abgeleiteten Kontextes sind.

Um Systemintegrität sicherzustellen, sollte dennoch aus Sicht einer Kontextinstanz, ein einzelnes Artefakt immer *entweder* als Kontextproduzent, *oder* als Kontextkonsument agieren. So können direkte Rekursionen vermieden werden, welche zu Problemen bei der algorithmischen Kontextverarbeitung führen können. Das hier entwickelte Kontextmanagementsystem stellt Mechanismen bereit, um dies zu gewährleisten.

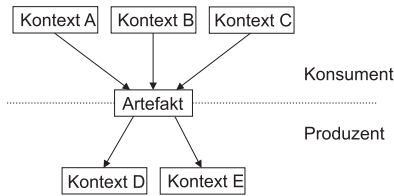


Abbildung 3.3. Produzent-Konsument-Schema: Artefaktsicht

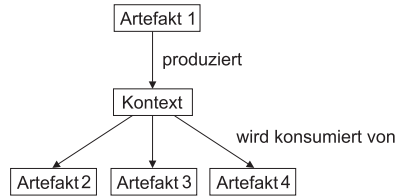


Abbildung 3.4. Produzent-Konsument-Schema: Kontextsicht

Abbildung 3.3 zeigt den Austausch von Kontextinformationen aus einer artefaktzentrierten Sicht. Das Artefakt agiert als Konsument und/oder Produzent eines oder mehrerer Kontexte. Dabei sollen konsumierte Kontextinstanzen stets von produzierten Kontextinstanzen unterscheidbar sein.

Abbildung 3.4 zeigt den Informationsaustausch zwischen Artefakten aus einer kontextzentrierten Sicht: Hier wird eine Kontextinstanz von einem bestimmten Artefakt produziert und dann von einem oder mehreren anderen Artefakten konsumiert.

Aus der Modellierung der Interaktion von Artefakten durch den Austausch von Kontexten in einem Produzent-Konsument-Schema folgt

Satz 3.1 *Jede Kontextinstanz hat genau einen Produzenten und kann $0, \dots, n \in \mathbb{N}$ Konsumenten haben.*

Die Modellierung der Interaktion zwischen Artefakten einer ubiquitären Informationsumgebung erlaubt das Aufstellen so genannter Verarbeitungsgraphen. Die Abbildungen 3.3 und 3.4 stellen einfache Beispiele solcher Verarbeitungsgraphen dar. Anhand des Verarbeitungsgraphen eines Kontextes lässt sich ablesen, welche alternativen Möglichkeiten in einer ubiquitären Informationsumgebung existieren, diesen Kontext zu aggregieren. Zusätzlich liefert der vollständige Verarbeitungsgraph einer ubiquitären Informationsumgebung durch die Anzahl seiner Knoten und Kanten ein Maß für deren Komplexität.

Das Produzent-Konsument-Schema dient der Entkoppelung von Kontextbereitstellung und Kontextverwendung. Dadurch sind Artefakte die Kontexte konsumieren potentiell unabhängig von solchen, die diese Kontexte produzieren. Allerdings muss auch das Kommunikationsmodell einer ubiquitären Informationsumgebung die Entkoppelung unterstützen.

3.6 Kontexthierarchie

Kontexte unterschiedlichen Abstraktionsgrades stellen die Kontextverarbeitung in ubiquitären Informationsumgebungen vor charakteristische Probleme. Auf den unteren Abstraktionsstufen besteht dieses Problem in der Handhabung von Ungenauigkeiten in den erfassten Sensordaten und hinterlegten Metainformationen (Auflösungsproblem). Auf höheren Ebenen müssen die Probleme der algorithmischen Zusammenführung multi-modaler Daten gelöst werden. Von zentralem Interesse für die vorliegende Arbeit sind dabei Lösungen für Probleme im Zusammenhang mit der Synthese von Kontexten auf der Basis abstrakter Kontexte als Ausgangsdaten.

Diese potentiellen Probleme der Verarbeitung und die charakteristischen Eigenschaften von Kontexten unterschiedlichen Abstraktionsgrades sollen bei der Verarbeitung der Kontexte beziehungsweise deren Generierung berücksichtigt werden. Es ist daher sinnvoll, Kontexte in ubiquitären Informationsumgebungen hierarchisch zu ordnen. Dabei bestimmt sich die Position einer einzelnen Kontextinstanz innerhalb der Hierarchie durch den Abstraktionsgrad der durch den Kontext repräsentierten Information. Je höher also ein Kontext in der Hierarchie steht, desto abstrakter ist die Information, die er repräsentiert.

3.6.1 Dynamisches Verarbeitungsmodell für Kontexte

Aus der Analyse von Kontexten in ubiquitären Informationsumgebungen lässt sich die Unterscheidung von drei grundlegenden Abstraktionsstufen von Informationen in einem kontextverarbeitenden System ableiten: Auf der untersten Ebene befinden sich Sensordaten und Metainformationen, da sie nur in soweit eine Abstraktion von den realen Bedingungen beinhalten, als diese durch Einschränkungen in der Auflösung im Falle von Sensordaten, oder durch eine klassifizierende Beschreibung im Falle von Metainformationen, zustande kommt.

Eine Abstraktionsstufe darüber befinden sich Kontexte, die direkt aus Sensordaten und Metainformationen abgeleitet werden. Diese Kontexte zeichnen sich dadurch aus, dass ihre Produzenten über umfassendes Wissen bezüglich der Qualität dieser Kontexte verfügen können. Alle Informationen, die zur Entstehung des Kontextes in einem Artefakt beitragen, werden von diesem selbst erfasst oder durch dieses bereitgestellt. Benötigte Sensoren sind Teil der Artefakthardware.

Ihre Eigenschaften, speziell solche, welche die Qualität des resultierenden Kontextes beeinflussen können, können bei der Erstellung der Anwendung, die auf einem Artefakt arbeitet, berücksichtigt werden. Ähnliches gilt für die, von einem Artefakt verwendeten Metainformationen, da diese im Artefakt selbst hinterlegt sind und somit der Kontrolle des Anwendungsentwicklers unterliegen.

Definition 3.13 (Kontext erster Ordnung) *Ein Kontext erster Ordnung wird direkt in einem einzigen Schritt aus Sensordaten und Metainformationen abgeleitet.*

Abbildung 3.5 zeigt die Einordnung von Kontexten anhand ihres Abstraktionsgrades in einer ubiquitären Informationsumgebung. Werden bei der Ableitung von Kontexten nicht nur Sensordaten und Metainformationen herangezogen, sondern abstrakte Informationen in Form anderer Kontexte weiter verarbeitet, so hat das verarbeitende Artefakt, respektive die verarbeitende Anwendung, nicht die Kontrolle und damit auch kein direktes Wissen über den Entstehungsprozess der verwendeten Daten.

Definition 3.14 (Kontext höherer Ordnung) *Kontexte höherer Ordnung werden abgeleitet von mindestens einem Kontext erster Ordnung und/oder einem Kontext höherer Ordnung und optional Sensordaten und Metainformationen.*

Kontexte höherer Ordnung stellen in großen ubiquitären Informationsumgebungen den Hauptteil der verfügbaren Kontextdaten dar. Alle Kontexte, die durch die modulare kompositorische Verarbeitung anderer Kontexte entstehen sind der Gruppe der Kontexte höherer Ordnung zuzuordnen. Damit ist es von zentralem Interesse speziell die Qualität dieser Kontexte höherer Ordnung in einer ubiquitären Informationsumgebung für die verarbeitenden Anwendungen beurteilbar zu machen und so die Qualität der auf ihnen basierenden Dienste zu sichern.

Die in Abbildung 3.5 dargestellte hierarchische Ordnung von Kontexten repräsentiert also nicht nur deren Abstraktionsgrad, sondern auch die Beurteilbarkeit der Qualität dieser Daten, wenn keine weiteren QoC-Mechanismen zum Einsatz kommen. Als Folge ergibt sich der zentrale

Satz 3.2 *Die Beurteilung und Handhabung der Qualität von Kontexten höherer Ordnung bedarf spezieller Verfahren, welche die Charakteristika der Verarbeitung höherer Kontexte in ubiquitären Informationsumgebungen und deren Eigenschaften berücksichtigen.*

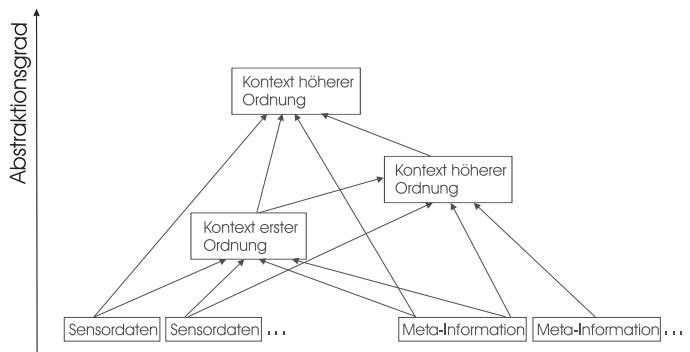


Abbildung 3.5. Hierarchische Anordnung von Kontexten im dynamischen Verarbeitungsmodell

Neben der Ordnung anhand des Abstraktionsgrades, existiert auf den Kontexten einer ubiquitären Informationsumgebung eine Ordnung der temporalen Entstehung der Kontexte (siehe Abbildung 3.6), die eng mit der hier eingeführten hierarchischen Ordnung verbunden ist.

Die temporale Ordnung von Kontexten induziert neben dem Zeitpunkt zu dem eine Information in der ubiquitären Informationsumgebung verfügbar wird, für jeden Kontext ein Alter. Das Alter ist eine der Grundeigenschaften von Kontexten, welche verwendet werden können, um die Qualität dieser zu bestimmen.

3.7 Entkoppelte Kommunikation: Kontext-Raum

In den vorangegangenen Abschnitten wurden Kontext und dessen Verarbeitung formalisiert. Ausgehend davon wird nun das Modell eines Kommunikationsraumes – des Kontext-Raumes – entwickelt. Dieses Modell der Kommunikation von Kontexten ermöglicht den entkoppelte Kontextaustausch zwischen Artefakten im Sinne des in Abschnitt 3.5 eingeführten Produzenten-Konsumenten-Schemas.

Grundlage des Kontext-Raumes (CSpace) ist der von Albrecht Schmidt et al. entwickelte FuzzySpace [147, 144]. Der FuzzySpace ist eine Erweiterung des Tupel-Raumes (Tuple Space) von Gelernter und Bernstein [60].

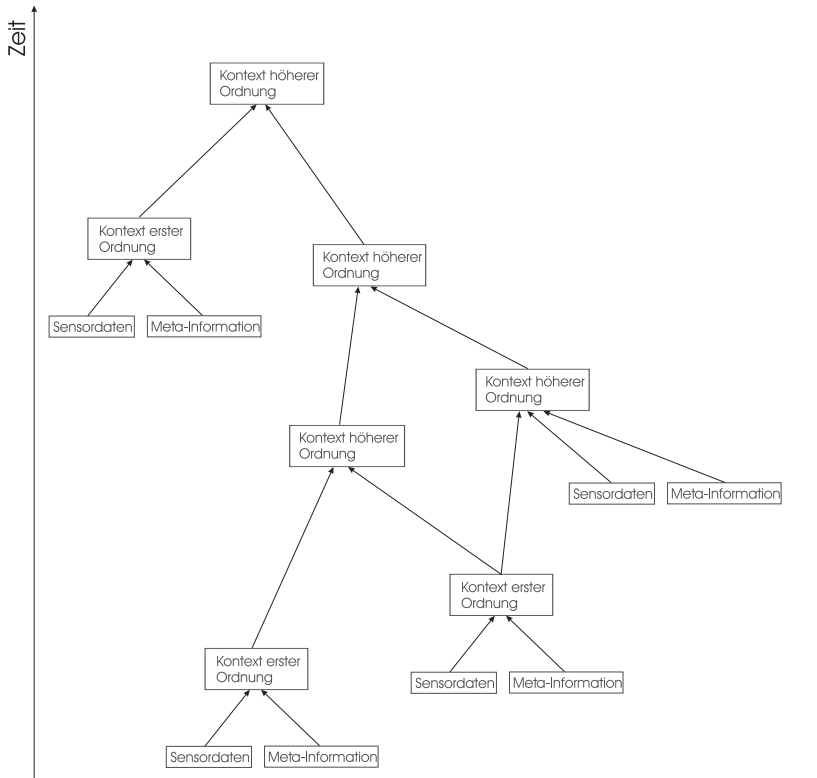


Abbildung 3.6. Temporale Entwicklung von Kontexten

Der CSpace modelliert die Kommunikation und Interaktion der Teilnehmer einer ubiquitären Informationsumgebung durch den Austausch von Kontexten.

Der CSpace unterstützt, wie der Tupel-Raum, die entkoppelte Kommunikation zwischen verschiedenen Teilnehmern des Kommunikationssystems. Entkopplung bedeutet dabei, dass Informationsquelle und Informationssenke, um Daten auszutauschen, keinerlei Informationen über ihre Kommunikationspartner benötigen (Abbildung 3.7). Eine klassische identitätsbezogene 1:1 Kommunikation durch Adressierung findet nicht statt. Statt dessen fügen Kontextproduzenten Kontext-Tupel $C = \{t, a, d\}$ in den CSpace ein, die von Kontextkonsumenten anhand von Selektionskriterien, in diesem Falle dem Kontexttyp, ausgewählt und gelesen werden können. Dadurch wird eine entkoppelte 1:n Kommunikation ermöglicht.

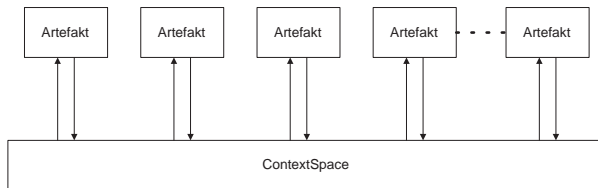


Abbildung 3.7. Entkoppelte Kommunikation: CSpace

3.7.1 Operatoren des ContextSpace

Für die Interaktion von Anwendungen mit dem CSpace stehen eine Reihe generischer Operatoren zur Verfügung.

- `cspace.create(channel_time)`

Der Operator `cspace.create` erzeugt einen CSpace. Da der CSpace ein globales Kommunikationsmodell innerhalb einer ubiquitären Informationsumgebung darstellt und somit immer nur eine Instanz zulässig ist, kann auf einen Identifikator verzichtet werden. Der Parameter `channel_time` gibt die maximale Zeit an, die eine Kontextinstanz im CSpace zur Verfügung steht. Wird `channel_time` nicht angegeben verbleiben alle Kontexte unbegrenzt im CSpace.

- `cspace.add(C)`

Mit dem Operator `cspace.add` kann ein Kontextproduzent ein Kontext-Tupel $C = \{t, a, d\}$ in den CSpace einfügen.

- `cspace.subscribe(t)`

Der Operator `cspace.subscribe` ermöglicht es einem Kontextkonsumenten für den Zugriff auf alle Kontexte C mit $C_t = t$ im CSpace zu registrieren. Der Subskriptionsoperator unterstützt damit die subskriptionsbasierte Kommunikation von Kontextkonsumenten und Produzenten über den CSpace. Die Subskription erfolgt dabei ausschließlich auf den Kontexttyp t . Ab dem Zeitpunkt der Subskription, stehen dem Kontextkonsumenten alle Kontexte vom Typ t , die sich im CSpace befinden zur Verfügung. Ein Konsument kann sich für beliebig viele Kontexte parallel registrieren.

- `cspace.read()`

Durch den Operator `cspace.read()` kann ein Konsument alle Kontexte C aus dem CSpace lesen, für deren Typen t er sich zuvor subskribiert hat. Alle subskribierten Kontexte werden dem Konsumenten dabei als Kopie in einer Liste bereitgestellt. Die Originalkontexte C selbst, verbleiben im CSpace und können weiterhin von anderen Konsumenten angefordert werden.

- `cspace.clear()`

Der Operator `cspace.clear` löscht den CSpace einer ubiquitären Informationsumgebung. Dabei gehen alle im CSpace vorhandenen Kontexte *C* verloren.

- `cspace.maintain()`

Durch den Operator `cspace.maintain` werden alle Kontexte *C* aus dem CSpace entfernt, die sich länger als die durch `channel_time` angegebene Zeitspanne im CSpace befinden.

Der CSpace unterstützt im Gegensatz zum FuzzySpace ausschließlich die direkte typbezogene Auswahl von Kontexten. Im FuzzySpace wird die Auswahl von Kontexten auf die temporalen und lokationsbezogenen Eigenschaften von Kontexten ausgedehnt [147]. Die Gruppierung von Kontexten erfolgt dabei anhand von Schablonen, welche unscharfe Mengen (engl. fuzzy sets) bezüglich der Relevanz der Kontexte beschreiben [144].

Die Beschränkung auf eine rein typbezogene Subskription auf Kontexte hat in der Praxis den Vorteil, dass der CSpace als abstraktes Modell der Kommunikation und Interaktion, keine verarbeitenden Funktionen bezüglich der über ihn ausgetauschten Kontexte implementieren muss. Im Gegensatz dazu muss der FuzzySpace die von Konsumenten angegebenen Schablonen als unscharfe Menge interpretieren und die Zugehörigkeit aller in ihm gespeicherten Kontexte zu diesen Mengen evaluieren.

Der CSpace stellt insofern ein allgemeineres Kommunikationsmodell als der FuzzySpace dar, da die Evaluierung weiterer Auswahlkriterien, wie die der Kontextqualität, anhand der verfügbaren Kontextattribute, dem Konsumenten des Kontextes überlassen wird. Dadurch wird größtmögliche Flexibilität der Anwendungen einer ubiquitären Informationsumgebung durch die individuelle Auswertung der Kontextattribute ermöglicht.

3.7.2 Einordnung des CSpace bezüglich des Kostenmodells

Der Kontext-Raum ist ein abstraktes Modell der Kommunikation in ubiquitären Informationsumgebungen. In Abschnitt 3.2 wurden die zu erwartenden Kosten der Kommunikation und Kontextverarbeitung solcher Modelle berechnet. Nun

wird das Modell des CSpace bezüglich des dort aufgestellten Kostenmodells analysiert und auf seine Eignung für den Einsatz in großen und hochdynamischen ubiquitären Informationsumgebungen geprüft.

Der CSpace unterstützt die entkoppelte Kommunikation von Kontextproduzenten und -konsumenten. Da der CSpace selbst keine Funktionen auf den kommunizierten Kontexten ausführt, sondern diese nur entgegennimmt und weitergibt, sind keine zentralen Komponenten notwendig. Die einzige Kontextauswahlmethode die der CSpace unterstützt, ist die Subskription von Konsumenten. Der CSpace ist also ein dezentrales subskriptionsbasiertes Kommunikationsmodell.

Die Kostenrechnung in Abschnitt 3.2 belegt, dass durch dezentral organisierte subskriptionsbasierte Kommunikationsmodelle in großen, hochdynamischen ubiquitären Informationsumgebungen die geringsten Kosten anfallen. Diese Kommunikationsmodelle sind besonders gut geeignet, da sie auch mit einer steigenden Anzahl von Netzwerksegmenten gut skalieren.

3.8 Attributierte Kontexte

Die Attributierung von Kontexten ist die Grundlage des in dieser Arbeit entwickelten Kontextmanagementsystems. In diesem Abschnitt wird geklärt, welche Vorteile die getrennte Betrachtung von Kontextattributen und semantischen Kontexten hat und wie auf der Basis der hier eingeführten Kontextattribute, ein Kontextmanagementsystem aufgebaut werden kann. Das Kontextmanagementsystem selbst hat das Ziel, die Qualität der Kontexterkennung auf semantischer Ebene zu verbessern.

Ausgangspunkt für die hier angestellten Betrachtungen sind aus der Literatur bekannte Ansätze zur Qualitätssicherung in der Kontextverarbeitung. Zu nennen sind hier unter anderem die Arbeiten von Gray und Salber [63] sowie Schmidt [144, 145] und Henricksen [78] bei denen die Beschreibung und Nutzung allgemeiner Kontexteigenschaften für die Optimierung der semantischen Verarbeitung im Vordergrund stehen. Aber auch Arbeiten von Buchholz et al. [24] und Krause und Hochstätter [104] und Strang et al. [156, 157], die sich hauptsächlich mit der Modellierung der Qualität von Kontextinformationen beschäftigen.

In diesen Arbeiten werden verschiedene Eigenschaften von Kontexten diskutiert, die zur Bestimmung der Kontextqualität herangezogen werden können oder diese im Bezug auf die Weiterverarbeitung von Kontexten beeinflussen:

- Vertrauenswürdigkeit und Präzision beschreiben die Divergenz zwischen Kontextaussage und realer Situation [50, 48, 29, 63, 29].
- Abdeckung oder Lokalität beschreiben Eigenschaften, die mit dem physikalischen Ort der Kontextverarbeitung assoziiert werden [50, 48, 63, 144, 145, 64, 103, 72].
- Auflösung beschreibt die Granularität, mit der Veränderungen der realen Situation in Kontexten widergespiegelt werden [50, 48, 63, 24, 79, 85, 72].
- Rechtzeitigkeit, Alter und Frequenz beschreiben temporale Eigenschaften von Kontexten [50, 48, 63, 144, 145, 143, 23, 58].

Dabei fällt auf, dass sich nur wenige der Arbeiten mit Kontextqualität aus Sicht des Systemmanagements auseinander setzen. Der Großteil der zitierten Arbeiten behandelt die angeführten Eigenschaften als Teil der semantischen Kontextmodellierung. Durch diese inhärente Verknüpfung von Bedeutung und Eigenschaften der Kontexte, wird die Entwicklung allgemeiner Managementmechanismen erschwert.

3.8.1 Dimensionen semantischer Kontexte

Wie bereits in Kapitel 2 erläutert wurde, stellen Kontexte eine Abstraktion der sensorischen Daten dar, die in einer ubiquitären Informationsumgebung erhoben werden. Dabei regeln semantische Kontextmodelle in welcher Weise, die durch die Kontexte bereitgestellten Beschreibungen der realen Welt, zu interpretieren sind. Semantische Kontextmodelle legen auch fest, wie die inhaltlichen Informationen eines Kontextes kodiert werden. Wie wir bereits gesehen haben existiert eine Vielzahl semantischer Kontextmodelle, die verschiedenen Anforderungen von Anwendungen gerecht werden. In der Vergangenheit wurden meist alle Aspekte des Kontextes in diese semantische Kodierung einbezogen, so dass nur Anwendungen und Artefakte, die mit dem gleichen Kontextmodell arbeiteten in der Lage waren, die Informationen zu verarbeiten. Dieser auf Anwendungsebene sinnvolle Ansatz bewirkt allerdings, dass sich keine allgemeineren, nicht von speziellen Anwendungen abhängigen Verwaltungsmechanismen für ubiquitäre Informationsumgebungen entwickeln ließen.

Eine genauere Analyse zeigt, dass sich neben den inhaltlichen anwendungsbezogenen Daten Eigenschaften der Kontexte extrahieren lassen, die von der inhaltlichen Bedeutung des Kontextes unabhängig sind: die Kontextattribute (vgl. Definition 3.7). Unabhängigkeit von der inhaltlichen Bedeutung heißt hierbei, dass jeder Kontext diese Eigenschaften hat, egal welche Bedeutung seinen Daten innerhalb einer ubiquitären Informationsumgebung zufällt. Es zeigt sich sogar, dass diese Kontextattribute unabhängig von dem gewählten semantischen Kontextmodell existieren.

Hinweise auf Kontextattribute, wie sie hier verstanden werden, finden sich in vielen wissenschaftlichen Arbeiten zum Thema Kontextverarbeitung. Bereits die ersten Arbeiten in diesem Themenbereich [143, 23, 58, 50] erwähnen bestimmte besondere Aspekte von Kontexten. Die in der Literatur am weitesten häufigsten genannten Kontexteigenschaften sind die Lokalität von Kontexten und ihr Alter. Ebenfalls oft genannt werden Verlässlichkeit und Auflösung der Kontextdaten.

In bisherigen Systemen werden die Kontextattribute immer ausschließlich als Teil des semantischen Kontextes modelliert. Daher stehen sie nicht unabhängig von dem zu Grunde liegenden semantischen Kontextmodell zur Verfügung, sondern sind untrennbar mit der inhaltlichen Bedeutung eines Kontextes verbunden.

Eigenschaften von Kontexten, wie die oben genannten, tragen zu der semantischen Interpretation eines Kontextes bei, da sie wichtige Aspekte des Kontextes und der Kontextgewinnung repräsentieren. Daher müssen sie Teil des semantischen Kontextes sein oder von der Instanz, die, innerhalb einer Anwendung semantische Kontexte verarbeitet, zugreifbar sein.

Neben dem Wert der Kontextattribute für die inhaltliche Verarbeitung, können sie als Basis für ein Kontextmanagementsystem dienen, das die Sicherung der Qualität von Kontexten auf Anwendungsebene erlaubt. Dabei werden, wie in Kapitel 2 beschrieben, die beschreibenden Informationen zu einem Kontext, hier die Kontextattribute, benutzt, um Kontexte, die den Qualitätsanforderungen einer Anwendung nicht genügen, bereits vor deren semantischer Verarbeitung, verwerfen zu können. Dazu müssen die Kontextattribute allerdings unabhängig von der semantischen Kontextverarbeitung bereits auf einer tieferen Schicht der Anwendungsarchitektur verfügbar sein. Kann dann durch das Kontextmanagementsystem entschieden werden, dass ein Kontext weiterverarbeitet werden soll, so müssen die Attributwerte der semantischen Kontextverarbeitung über geeignete Schnittstellen zur Verfügung gestellt werden, da sie wie bereits oben erwähnt, auch als semantischer Aspekt in die Verarbeitung der Kontexte eingehen.

Auf der Ebene des Kontextmanagementsystems bilden die Attribute eines Kontextes die Dimensionen eines Informationsraumes innerhalb dessen die Bewertung der Qualität aus Sicht der Anwendung stattfindet. Dabei beschreibt jede Dimension eine charakteristische Eigenschaft von Kontexten, die zu deren Klassifizierung herangezogen werden kann.

Im Folgenden werden die für das hier eingeführte System definierten Klassifikationsdimensionen des Informationsraumes und die damit verbundenen Kontextattribute eingeführt. Das hier vorgestellte Modell attributierter Kontexte dient nicht zur vollständigen Beschreibung aller von semantischen Interpretationen unabhängiger Kontexteigenschaften. Vielmehr wurden die Attribute gezielt ausgewählt, um Aspekte der QoC evaluieren zu können, die in realen ubiquitären Informationsumgebungen bei der Verarbeitung von Kontexten zu Problemen führen. Dieser Auswahl liegt eine langjährige Erfahrung mit der Entwicklung und dem Betrieb ubiquitärer Informationsumgebungen sowie die Analyse gängiger Anwendungs- und Problemszenarien aus der Literatur [44, 54, 77] zu Grunde.

3.8.2 Temporale Dimension

Kontext beschreibt eine Situation in der realen Welt und macht sie so für Rechensysteme zugänglich [48]. Dabei ist der Kontext ein Schnappschuss eines bestimmten Ausschnittes der Gesamtsituation zu einem bestimmten Zeitpunkt. Die reale Welt selbst und damit die Situation, die der Kontext wiederzugeben versucht, ist ständiger Veränderung unterworfen. Das Erfassen dieser Veränderung, also der Dynamik der Umwelt, ist für die meisten Anwendungen in ubiquitären Informationsumgebungen von größerer Bedeutung als eine einzelne Momentaufnahme [146, 64].

Betrachtet man Kontext, wie hier, als Daten, die in einer ubiquitären Informationsumgebung verarbeitet werden und nicht synonym mit der Situation, die er beschreiben soll, so erfährt eine Kontextinstanz in dem hier vorgestellten Modell nach ihrer Erzeugung keinerlei Veränderung. Die Dynamik der Umgebung wird durch die Erzeugung immer neuer Kontextinstanzen einer Klasse repräsentiert, wobei sich die Werte der Attribute und Kontextdaten entsprechend der Veränderungen in der Umgebung ändern.

Somit ist jedem Kontext ein Erzeugungszeitpunkt zuzuordnen. Zu diesem Zeitpunkt ist der erzeugte Kontext die beste Beschreibung der Umgebung, welche die erzeugende Anwendung, der Kontextproduzent, zur Verfügung stellen kann.

Ist der Kontext erzeugt und finden Veränderungen in der Umgebung statt, ist nicht mehr sicher, inwieweit die Kontextdaten die aktuelle, nun eventuell veränderte Situation repräsentieren. Erste Überlegungen zu der möglichen Formalisierung temporaler Einflüsse auf Kontextdaten finden sich in den Arbeiten von Albrecht Schmidt [147, 144].

Schmidt führt den Begriff der Relevanz von Kontexten ein und meint damit ein abstraktes Maß für den Nutzen, den eine kontextsensitive Anwendung aus einem Kontext ziehen kann. Er geht in seinen Überlegungen davon aus, dass die Relevanz eines Kontextes in seiner zeitlichen Dimension durch eine monoton fallende Funktion beschrieben werden kann. Dabei stellt diese Funktion, welche die temporale Degression des Kontextes beschreibt, aus seiner Sicht selbst eine Eigenschaft des Kontextes (der Kontextklasse) dar. Das bedeutet in Schmidts Modell kann einem Kontext bei seiner Erzeugung aufgrund seines Typs eine monoton fallende Funktion zugeordnet werden, welche die zeitliche Veränderung seiner Relevanz eindeutig beschreibt.

Die Erfahrung mit Anwendungen in realen ubiquitären Informationsumgebungen wie dem AwareOffice [190] zeigen allerdings, dass eine Bewertung von Kontextdaten in der temporalen Dimension ein differenzierteres Vorgehen erfordert. Wenn die Anwendungen einer ubiquitären Informationsumgebung nicht alle aus einer Quelle stammen, kann zum Beispiel nicht generell ausgeschlossen werden, dass Anwendungen existieren, die sich primär für historische Daten interessieren, die also bereits ein gewisses Alter erreicht haben. Auch dieser Fall könnte in Schmidts Modell abgedeckt werden, in dem man den Relevanzbegriff aufweicht. Eine Anwendung mit Interesse an historischen Kontexten müsste dann auch ein bestimmtes Intervall der Relevanzwerte filtern. Allerdings müssen Konsumenten in diesem Modell zwingend über Wissen von Details der Implementierung der Produzenten von Kontexten verfügen, da die Produzenten der Kontexte die Relevanzfunktion vorgeben. Dies kann in großen ubiquitären Informationsumgebungen aufgrund der zu erwartenden Heterogenität nicht durchgesetzt werden.

Weiter kann nicht davon ausgegangen werden, dass sich die Entwicklung der temporal bestimmten Relevanz für alle Konsumenten ähnlich oder gar gleich verhält. Somit muss angenommen werden, dass nur der Konsument selbst über genug Wissen, die Anforderungen seiner Anwendung betreffend, verfügt, um für Kontextdaten, die er verarbeiten möchte, eine temporale Relevanzfunktion aufzustellen. Dafür genügt es, dass der Zeitpunkt der Erzeugung eines Kontextes als Attribut mitgeteilt wird.

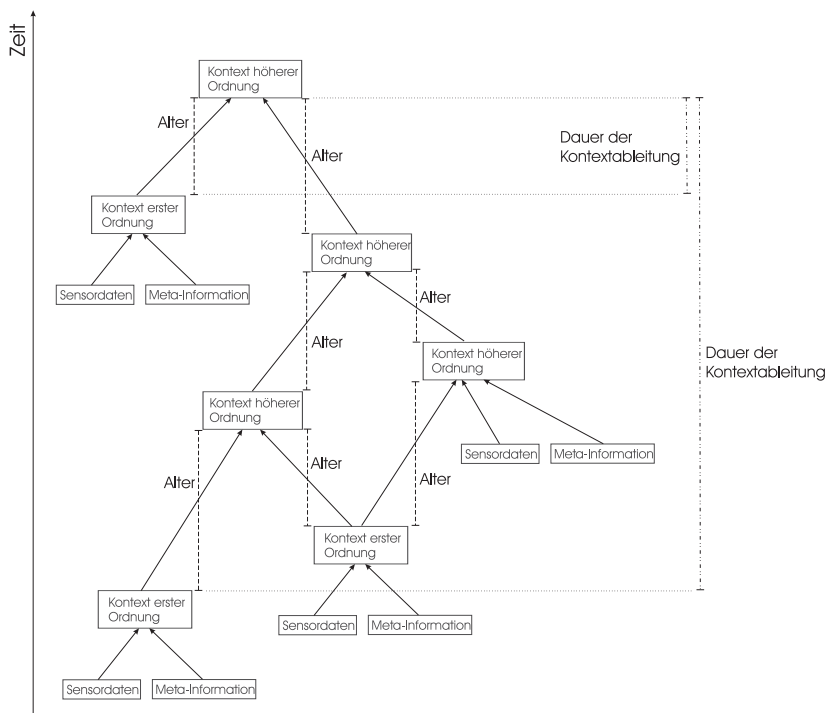


Abbildung 3.8. Alter und Ableitungsdauer von Kontexten

Neben dem Alter existiert in der zeitlichen Dimension von Kontextdaten eine weitere wichtige Kenngröße, die zur Beurteilung der Relevanz herangezogen werden muss. Dabei handelt es sich um die Dauer der Kontextableitung. Abbildung 3.8 zeigt das Alter von Kontexten und die Dauer der Ableitung eines Kontextes im Vergleich. Das Alter eines Kontextes ist, wie bereits beschrieben, die Zeitspanne von dessen Erzeugung bis zu seiner Verarbeitung, oder genauer bis zur Überprüfung des entsprechenden Attributwertes. Die Dauer der Ableitung, die in der Abbildung zur Übersichtlichkeit nur für den zeitlich letzten Kontext aufgetragen ist, entspricht der Zeit von der Erzeugung des ersten Kontextes, der Teil des Ableitungsgraphen eines Zielkontextes ist, bis zu dessen Verarbeitung.

Da die Erzeugung von Kontexten den in Abschnitt 3.6 eingeführten kausalen und temporalen Ordnungen folgt, reicht es bei der Verarbeitung von Kontexten höherer Ordnung nicht aus, nur das Alter der Quellkontexte zu berücksichtigen. Kontexte erster Ordnung können in grober Näherung, wie in dem Modell von Schmidt, als Momentaufnahmen der Situation in der Umgebung verstanden werden. Im Gegensatz dazu repräsentieren Kontexte höherer Ordnung nicht generell das Bild einer Umgebung zu einem bestimmten Zeitpunkt. Durch die temporale Abhängigkeit der Ableitung fassen diese Kontexte im allgemeinen Fall Beschreibungen der Umgebung über einen gewissen Zeitraum, die Dauer der Ableitung, zusammen.

Aus Sicht einer konsumierenden Anwendung kann die Qualität eines Kontextes nicht nur von dessen Alter abhängen, sondern auch von der Länge des Ableitungsintervalls, in dem der Kontext entstanden ist. Man kann annehmen, dass eine kürzere Ableitungsdauer zu qualitativ hochwertigeren Kontexten führt. Allerdings kann auch in diesem Fall nicht ausgeschlossen werden, dass bestimmte Anwendungen in einer heterogenen Umgebung gerade an Kontexten interessiert sind, welche längere Zeitspannen oder sogar temporale Abfolgen von Ereignissen, repräsentieren.

3.8.3 Räumliche Dimension

Sowohl Kontext als Beschreibung, als auch die Situation, die durch Kontext beschrieben wird, beziehen sich auf einen bestimmten Ort oder eine Region. Räumliche Konzepte sind für den Menschen bei der Bewältigung seiner alltäglichen Aufgaben von grundlegender Wichtigkeit. David Kirsh hat die Zusammenhänge von menschlicher Interaktion und räumlichen Konzepten bereits vor 10 Jahren

erkannt und eingehend untersucht [99, 101, 100, 98]. Er kommt zu dem Schluss, dass der Mensch bei der Bewältigung alltäglicher Aufgaben extensiven Gebrauch von einer aktiven räumlichen Strukturierung seiner Umwelt macht, um die Komplexität der Aufgaben zu reduzieren und so seine Lösungseffizienz zu steigern.

Das Ziel der Kontextnutzung in ubiquitären Informationsumgebungen ist, wie in Kapitel 2 dargelegt wurde, computerisierte Artefakte zu schaffen, die den Benutzer durch ihr situatives Wissen unterstützen können. Dabei steht oft die Anpassung des Geräteverhaltens an die aktuelle Situation im Vordergrund. Setzt man dieses Ziel in Beziehung zu den Forschungsergebnissen von Kirsh, folgt daraus direkt die Wichtigkeit räumlicher Konzepte für die Kontextverarbeitung [14, 103, 72]. Daher kommt der Lokalität als Eigenschaft von Kontexten besondere Bedeutung bei der Bewertung der QoC zu.

Ein Kontext wird durch ein mobiles Artefakt produziert, das sich an einem bestimmten Ort befindet oder von einem Infrastrukturobjekt bereitgestellt wird, das einem räumlichen Bereich oder Ort zugeordnet ist. Dadurch besitzt jeder Kontext automatisch einen Entstehungsort bezüglich des räumlichen Systems, in dem eine ubiquitäre Informationsumgebung etabliert wird.

Ein Anwendungsbeispiel soll die Bedeutung der räumlichen Herkunft von Kontexten für deren Weiterverarbeitung veranschaulichen: Das Gebäude einer Firma besteht aus mehreren Räumen, darunter zwei Besprechungsräume, die mit CSCW Technologien ausgestattet sind. In den Besprechungsräumen befindet sich jeweils eine Wandtafel auf der mit speziellen Stiften (Boardmarkern) geschrieben werden kann. Um den Umgang mit den Tafelaufschrieben zu vereinfachen, sind in den Besprechungsräumen Digitalkameras installiert, welche Bilder von den Wandtafeln machen können. Die Kameras verarbeiten Kontexte, die von den Boardmarkern produziert werden. Sie machen ein Bild von der Wandtafel, wenn eine gewisse Menge neuer Text auf die Tafel geschrieben wurde. Die Bilder können dann archiviert oder den Teilnehmern einer Besprechung per Email zugesandt werden.

In diesem Szenario ist es sinnvoll, wenn die Kameras nur Kontexte von Stiften verarbeiten, die sich im gleichen Raum befinden. Die Kontexte der Stifte aus dem zweiten Besprechungsraum sind für die Kameraanwendung nicht relevant.

Hier zeigt sich, dass für viele Anwendungen die logische Kolo-kation von Kontexten von größerer Bedeutung ist, als der geometrische Abstand zu einer Kontextquelle. Zum Beispiel sind Anwendungen wie eine dynamische Anwesenheitstafel im Büroumfeld denkbar, die ausschließlich Kontexte verarbeiten, die in Mit-

arbeiterbüros erzeugt wurden. Findet eine solche logische Gruppierung von Kontexten bezüglich ihres Entstehungsortes statt, kann nicht mehr davon ausgegangen werden, dass die Relevanz von Kontexten für den Konsumenten mit dem Abstand zum Produzenten monoton fällt.

In vielen Anwendungen aus dem Bereich der lokationsbasierten Dienste (engl. location based services, LBS) ist der Entstehungsort nicht nur Kontexteigenschaft, sondern zugleich semantische Kontextinformation. Damit stellen LBS ein Beispiel für Gleichwertigkeit von semantischer Kontextinformation und Kontextattributen auf Anwendungsebene dar.

Analog zu seinem temporalen Relevanzmodell schlägt Schmidt ein Modell der räumlichen Relevanz vor [147, 144]. Er beschränkt sich hierbei auf ein zweidimensionales Abstandsmodell, in dem mit dem Abstand zum Erzeugungsort monoton fallende Relevanzfunktionen definiert werden. Der Autor geht davon aus, dass die Relevanz eines Kontextes generell mit dem Abstand zu seiner Quelle fällt. Die Relevanzfunktion wird vom Produzenten des Kontextes vorgegeben. Auch hier ergibt sich, wie schon in der temporalen Dimension, das Problem der Unabhängigkeit von Produzenten und Konsumenten in einer ubiquitären Informationsumgebung. Verfügen die Konsumenten nicht über das Wissen, wie der Produzent eines Kontextes die Relevanzfunktion anlegt, können sie diese nicht im Sinne eines Qualitätsmaßes benutzen, um Informationen zu filtern.

Wird dagegen der Entstehungsort eines Kontextes als Kontextattribut bereitgestellt, reicht es aus, wenn der Konsument mit dem gleichen Ortsmodell arbeitet, das der Produzent benutzt, oder in der Lage ist, die Ortsangabe in seinem Ortsmodell zu interpretieren. Diese Forderung entspricht der einer gemeinsamen Zeitdarstellung im temporalen Falle. Dann ist der Konsument eines Kontextes in der Lage dessen räumliche Relevanz, den Anforderungen seiner Anwendung entsprechend zu evaluieren, ohne zusätzliches Wissen über den Produzenten zu benötigen.

3.8.4 Verlässlichkeit von Kontext als Dimension des Informationsraumes

Kontexte stellen eine Abstraktion der realen Welt dar. Daher können und sollen sie diese nicht in ihrer gesamten Informationstiefe wiedergeben. Aus diesem Umstand ergeben sich einige wichtige Rahmenbedingungen, die bei der Verarbeitung von Kontexten beachtet werden müssen. In der Literatur werden diese meist unter dem Begriff der *Zuverlässigkeit von Kontexten* zusammengefasst.

Zuverlässigkeit von Kontexten subsumiert verschiedene Parameter von Kontext, welche die inhaltliche, also die semantische Qualität der Kontextinformation beschreiben [56, 53]. Die in der Literatur am häufigsten untersuchten Parameter dieser Kategorie sind

Auflösung [63, 24, 79, 85, 72] Unter der Auflösung von Kontexten wird die kleinste Informationsdifferenz verstanden, die sie darstellen können. Bei sensornahen Kontexten geringer Abstraktion kann es sich dabei um die kleinste physikalische Einheit handeln, die ein zu Grunde liegender Sensor erfassen kann. Im Falle abstrakter Kontexte gibt die Auflösung das Verhältnis zwischen dem Facettenreichtum der zu beschreibenden Situation und der Anzahl der Facetten an, die durch verschiedene Instanzen des Kontextes unterschieden werden können.

Genauigkeit [63, 29, 104, 133] Die Genauigkeit eines Kontextes sagt aus, um wie viele Einheiten seiner maximalen Auflösung ein Kontext im Durchschnitt von der Realität abweicht.

Eindeutigkeit [145, 51, 74] Da der Begriff der Auflösung für abstrakte Kontexte nicht ausreicht, um die Möglichkeit zu beschreiben, dass eine Kontextinstanz einer Situation nicht zweifelsfrei zugeordnet werden kann, wurde der Begriff der Eindeutigkeit eingeführt.

Richtigkeit [63, 145, 24, 185] Die Richtigkeit ist ein Maß für Fehler in der Aggregation oder Fusion von Kontexten, die zu einer generell fehlerhaften Situationsbeschreibung durch den Kontext führen.

Als übergeordnetes Maß der Zuverlässigkeit kommt dabei meist die Kontexterkennungsrate zum Einsatz.

Definition 3.15 (Kontexterkennungsrate) *Die Kontexterkennungsrate gibt das durchschnittliche Verhältnis zwischen korrekt erkannten Kontexten und der Anzahl der real vorhandenen, verschiedenen Kontexte an.*

Bei Verwendung der Kontexterkennungsrate, wie auch der obigen Kontextparameter, ist zu beachten, dass diese eine statistische Beschreibung des Kontextableitungsprozesses darstellen. Sie sind damit ungeeignet, die semantische Qualität einer einzelnen Kontextinstanz zu erfassen. In Abgrenzung zum Begriff der *Zuverlässigkeit* wird daher in dieser Arbeit der Begriff der *Verlässlichkeit* von Kontexten eingeführt.

Definition 3.16 (Verlässlichkeit von Kontext) *Die Verlässlichkeit von Kontext stellt ein Maß für die inhaltliche Qualität einer Kontextinstanz dar. Sie ist somit ein Kontextattribut, da sie jedem Kontext, unabhängig von seiner semantischen Bedeutung, ein Maß seiner inhaltlichen Qualität zuordnet.*

Dieses neue Modell der Verlässlichkeit von Kontexten ermöglicht die Erfassung und Berücksichtigung von Schwankungen in der Erkennungsqualität von Kontextalgorithmen auf Kontextinstanzebene. Im Gegensatz zu früheren Ansätzen [63, 24, 145, 74, 79, 85, 104, 133, 72] wird es dadurch für den Konsumenten einer Kontextinstanz möglich, anhand ihrer Verlässlichkeit zu entscheiden, ob ihre inhaltliche Qualität den Anforderungen seiner Anwendung genügt. Ist dem nicht so, kann der Kontext sofort verworfen werden, ohne dass eine Verarbeitung durch die semantischen Kontextalgorithmen erfolgen muss.

Können Kontexte niedriger Verlässlichkeit in einer ubiquitären Informationsumgebung nicht identifiziert und aus dem Verarbeitungsprozess entfernt werden, pflanzen sich Fehler in der Kontexterkenkung fort. Kontexte bilden die Basis für Entscheidungen innerhalb kontextsensitiver Anwendungen. Entspricht die Aussage eines Kontextes nicht der tatsächlichen Situation, ist also die Kontexterkenkung fehlergeschlagen, muss vermieden werden, dass dieser Kontext weiterverarbeitet wird und damit Basis weiterer Fehlentscheidungen in Anwendungen werden kann.

Modell der Verlässlichkeit von Kontexten

Der Verlässlichkeit, als Maß für die inhaltliche Qualität einer Kontextinstanz, liegt ein Verlaufsmodell zu Grunde. Ausgehend von der Annahme eines idealen Kontexterkennungsalgorithmus wird dadurch der Verlauf des Verlässlichkeitsmaßes bei einem Kontextwechsel in der realen Welt beschrieben.

Ein idealer Kontexterkennungsalgorithmus zeichnet sich dadurch aus, dass er reale Kontexte sicher erkennt. Das bedeutet, bei einem Kontextwechsel zwischen Kontexten, die der Erkenner erkennen kann, fluktuiert seine Ausgabe nicht zwischen verschiedenen Zuständen und nimmt auch keine anderen Werte als den Ausgangs- und Zielwert an. Sie wechselt lediglich ein einziges Mal vom Ausgangskontext auf den Zielkontext.

Abbildung 3.9 zeigt den zeitlichen Verlauf $C_{real}(t)$ eines möglichen Kontextwechsels von Kontext A nach Kontext B in einer realen Umgebung. Dabei ist

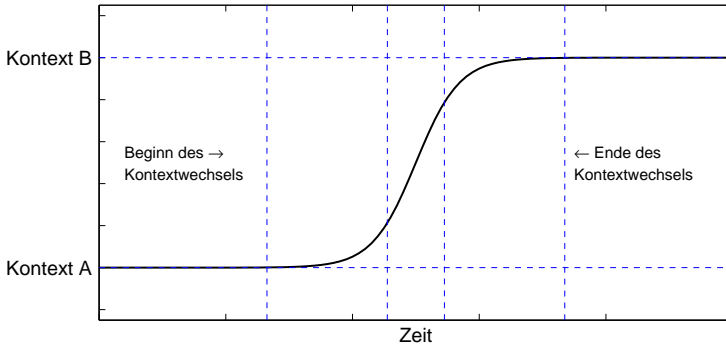


Abbildung 3.9. Zeitlicher Wechsel: Kontext A → Kontext B

zu beachten, dass ein Wechsel zwischen Kontexten in der realen Welt nie abrupt stattfindet, sondern ein kontinuierlicher Vorgang mit einer unendlichen Anzahl Zwischenzustände infinitesimalen Abstandes ist. Er kann daher durch eine stetige Funktion dargestellt werden.

Die Ausgabe eines idealen Kontextalgorithmus $C_{recog}(t)$, der den Kontextwechsel in Abbildung 3.9 erkennt, ist in Abbildung 3.10 dargestellt. Der ideale Erkennen ignoriert alle Zwischenzustände des Übergangs von Kontext A nach Kontext B. Er gibt kontinuierlich den Kontext aus, welcher die reale Situation am besten beschreibt.

Ein Maß für die Verlässlichkeit $\psi(C, t)$ des erkannten Kontextes ergibt sich als Funktion des Abstandes zwischen diesem und der realen Situation:

$$\psi(C, t) = f(|C_{real}(t) - C_{recog}(t)|) \quad (3.2)$$

Stimmen reale Situation und erkannter Kontext weitgehend überein, ist die Verlässlichkeit des erkannten Kontextes hoch, vergrößert sich jedoch der Abstand, so sinkt die Verlässlichkeit.

Abbildung 3.11 zeigt das hier entwickelte Verlaufsmodell der Verlässlichkeit von Kontext.

Aufgrund der Annahme eines idealen Erkenners ist die Verlässlichkeit des Kontextes (Kontext A) am Anfang des Beobachtungszeitraumes maximal. Mit Beginn

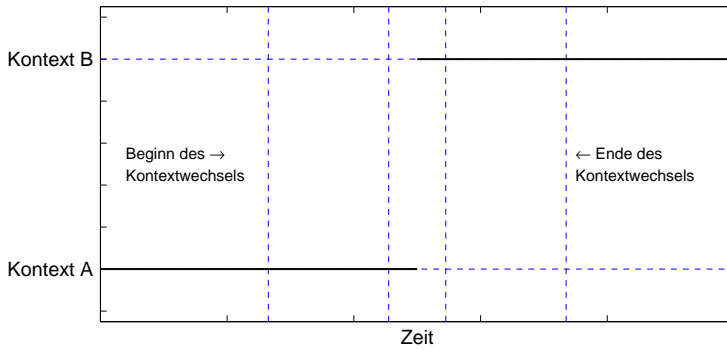


Abbildung 3.10. Idealer Erkennen: Wechsel Kontext A → Kontext B

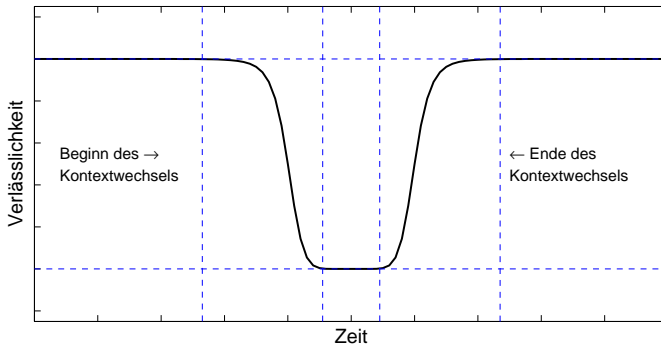


Abbildung 3.11. Verlaufsmodell der Verlässlichkeit von Kontext

des Kontextüberganges nimmt die Verlässlichkeit des Kontextes A ab bis sie ein Minimum erreicht. Aus Abbildung 3.9 ist zu ersehen, dass sich die reale Situation zu diesem Zeitpunkt zwischen Kontext A und Kontext B bewegt. In diesem Bereich minimaler Verlässlichkeit findet der Kontextwechsel $A \rightarrow B$ des Erkenners statt. Die Verlässlichkeit des Kontextes B steigt dann wieder an, da sich die reale Situation dem Kontext immer mehr annähert. Ist der Kontextübergang in der realen Welt abgeschlossen, erreicht der erkannte Kontext B maximale Verlässlichkeit.

Dieses Modell der Verlässlichkeit von Kontext dient dazu, Kontextkonsumenten Zugriff auf ein Kontextattribut zu geben, dass die inhaltliche Qualität der vorliegenden Kontextinstanz beschreibt. Der Konsument wird dadurch in die Lage versetzt, vor der semantischen Verarbeitung eines Kontextes anhand dieses Maßes zu entscheiden, ob die inhaltliche Qualität des Kontextes den Anforderungen der Anwendung entspricht.

Hervorzuheben ist, dass die hier definierte Verlässlichkeit, obwohl sie ein Maß für die inhaltliche Qualität des Kontextes darstellt, unabhängig von seiner semantischen Bedeutung ist. Im Gegensatz zu statistischen Maßen beschreibt sie nicht Eigenschaften des Kontexterkennungsprozesses, sondern eine Eigenschaft – ein Attribut – jeder individuellen Kontextinstanz.

3.8.5 Abhängigkeit zwischen Kontexten als Dimension des Informationsraumes

Kontext existiert in einer ubiquitären Informationsumgebung nicht isoliert. Artefakte erfassen Kontexte erster Ordnung durch Umweltsensoren und stellen diese anderen Konsumenten zur Verfügung. Diese nehmen die Kontexte auf und verarbeiten sie zu abstrakteren Kontexten höherer Ordnung weiter (siehe Abschnitt 3.6). Dabei entsteht ein Netz von Verarbeitungsprozessen, das die Beziehungen zwischen den, in einer ubiquitären Informationsumgebung verfügbaren Kontexten beschreibt: der Verarbeitungsgraph.

Durch das Ableiten entstehen informationelle Abhängigkeiten zwischen den Kontexten einer ubiquitären Informationsumgebung. Bei der Aggregation oder Fusion neuer Kontexte bleiben Teile der Information der Quellkontexte erhalten und gehen auf den Zielkontext über. Die informationelle Abhängigkeit ist eine Grundeigenschaft von Kontexten. Sie kann, wenn sie bei der Verarbeitung nicht beachtet wird, die Qualität der Kontexte negativ beeinflussen.

In der Literatur wurden Kontextabhängigkeiten bisher kaum untersucht. Lediglich die Einflüsse der Verarbeitung von nicht unabhängigen Daten auf bestimmte Algorithmen, die auch in der Kontexterkenkung angewendet werden, ist bekannt: Mustererkennung [38], Markov Ketten [12], Bayes'sche Netzwerke [116], statistisch-relationales Lernen (SRL) [114, 122], allgemeine Klassifikatoren [167], Regressionsmodelle [90, 132].

Im Allgemeinen wird bei der Anwendung von Algorithmen, wie Kalman Filtern [92], Markov Modellen [130, 105], Bayes'schen Netzwerken [59] oder ID3 und C4.5 [129] vorausgesetzt, dass die verarbeiteten Daten unabhängig sind oder höchstens ein gewisses Maß an Abhängigkeit enthalten. Dies kann allerdings in ubiquitären Informationsumgebungen nicht generell sichergestellt werden. Daher ist die Voraussetzung für den erfolgreichen Einsatz dieser Algorithmen das Vorhandensein eines Maßes für den Grad der informationelle Abhängigkeit von beliebigen Kontexten innerhalb einer ubiquitären Informationsumgebung.

Das hier entwickelte Modell führt als Maß der Abhängigkeit die *Verwandtschaft von Kontexten* ein.

Definition 3.17 (Verwandtschaft von Kontexten) *Die Verwandtschaft von Kontexten ist ein Maß für die informationelle Abhängigkeit von Kontextinformationen, die durch den Prozess der Ableitung von Kontexten höherer Ordnung entsteht.*

Kontextabhängigkeiten

Kontextabhängigkeiten treten in ubiquitären Informationsumgebungen in zwei Grundmustern auf, die durch die Analyse mehrerer Anwendungsszenarien [44, 54, 77] und die Simulation großer ubiquitärer Informationsumgebungen identifiziert werden konnten. Diese beiden Muster reichen aus, um die in ubiquitären Informationsumgebungen vorkommenden, nicht linearen Abhängigkeiten, zu beschreiben. Sie dienen daher in dieser Arbeit als Grundlage für die Entwicklung eines algorithmischen Systems zur quantitativen Beschreibung der Verwandtschaft von Kontexten.

Das erste Grundmuster, *Auffächerung und Multiplikation*, kann in ubiquitären Informationsumgebungen immer dann auftreten, wenn mehrere gleichartige Artefakte, die als Produzenten, Kontextinstanzen des gleichen Typs erzeugen, dieselben Kontexte konsumieren und als Basis für ihre Berechnungen verwenden.

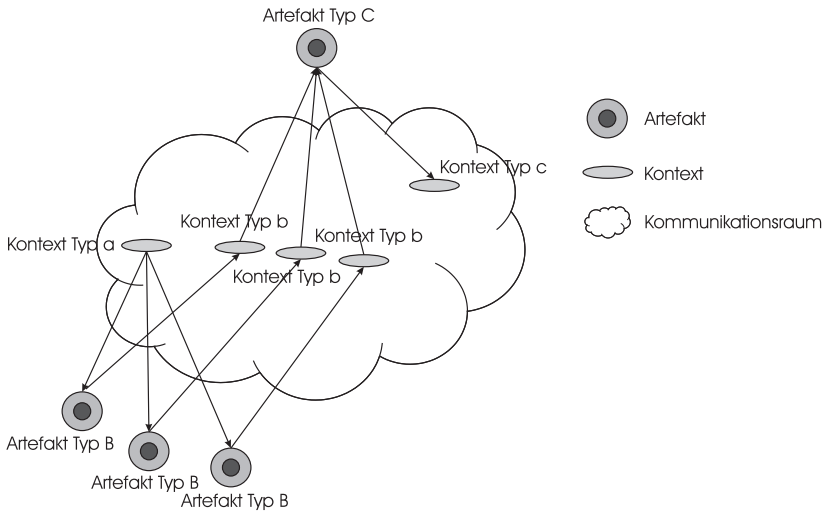


Abbildung 3.12. Auffächerung und Multiplikation

Abbildung 3.12 zeigt eine schematische Darstellung der Auffächerung und Multiplikation von Kontexten. Ein Kontext des Typs *a* wird parallel von mehreren Artefakten des Typs *B* konsumiert. Jedes der Artefakte *B* produziert auf der Basis des konsumierten Kontextes einen neuen Kontext von Typ *b*. Will nun ein Artefakt *C* die Kontexte vom Typ *b* konsumieren, um sie zu aggregieren, kann es abhängig von dem verwendeten Aggregationsalgorithmus notwendig sein, den Grad der Verwandtschaft der Kontexte *b* zu kennen.

Ein bekanntes Anwendungsbeispiel aus dem Bereich der besprechungsunterstützenden Anwendungen soll das Problem illustrieren: Um sich an eine Besprechungssituation anpassen zu können, werden persönliche Geräte wie Mobiltelefone und persönliche digitale Assistenten (PDAs) mit kontextsensitiven Anwendungen ausgestattet, die es ihnen erlauben, ihre Alarmierungsfunktion der Situation anzupassen. Dies kann zum Beispiel bedeuten, dass der Ruftton eines Mobiltelefons oder der Erinnerungsalarm eines PDA lautlos geschaltet werden, wenn sich diese Geräte in einer Besprechung befinden. Leiten nun die verschiedenen persönlichen Geräte in einer Besprechung den Besprechungskontext von denselben Quellkontexten ab, so sind die entstehenden (Besprechungs-) Kontext-

instanzen miteinander verwandt. Das bedeutet, die Information, die durch die Besprechungskontexte repräsentiert wird ist teilweise redundant, da sie nicht auf der Basis unabhängiger Quellen entstanden ist. Eine Anwendung, die Besprechungskontexte weiterverarbeiten kann, diese also konsumiert, muss in der dargestellten Situation entscheiden, wie sie mit diesen Kontexten umgeht.

In ubiquitären Informationsumgebungen, in denen Anwendungen nicht in der Lage sind, dynamisch während der Laufzeit auf Informationen über die Verwandtschaft von Kontextdaten zurückzugreifen, ist der Softwareentwickler gezwungen, bereits während der Erstellung der Anwendung zu entscheiden, wie diese in einer Situation, in der mehrere Kontextinstanzen eines Typs zur Verfügung stehen, mit der Verarbeitung fortfährt.

Ist eine Anwendung nicht in der Lage, dynamisch den Verwandtschaftsgrad von Kontexten zu berücksichtigen, sind vier verschiedene Vorgehensweisen denkbar:

- Die Anwendung wählt zufällig *einen* der zur Verfügung stehenden Kontexte aus. Zum Beispiel den, der als erster empfangen wurde.
- Die Anwendung wählt anhand eines Qualitätsmaßes, das nicht auf die Verwandtschaft der Kontexte Bezug nimmt, *einen* der Kontexte aus.
- Die Anwendung wählt *alle* Kontexte aus und verarbeitet sie ohne Rücksicht auf ihre eventuelle Verwandtschaft mit einem festen Algorithmus.
- Die Anwendung verarbeitet eine *Auswahl* von Kontexten, die anhand eines Qualitätsmaßes, das nicht auf die Verwandtschaft der Kontexte Bezug nimmt, getroffen wurde.

In all diesen Fällen bleibt der Verwandtschaftsgrad der Kontexte unberücksichtigt. Dies führt dazu, dass entweder Ableitungsalgorithmen, die unabhängige Eingangsdaten voraussetzen, nicht zum Einsatz kommen können, oder dass die Fehler, die durch abhängige Daten verursacht werden, in Kauf genommen werden müssen. Werden Fehler in Kauf genommen, führt dies wiederum zu einer sinkenden Kontexterkennungsrate dieser Anwendung und aller Anwendungen einer ubiquitären Informationsumgebung, die im Verarbeitungsgraphen nach dieser Anwendung die Kontexte verarbeiten, da sich der Fehler über die Ableitungskette fortpflanzt.

Kann eine Anwendung den Verwandtschaftsgrad von Kontextdaten bei der Verarbeitung berücksichtigen und ist dieser relevant für die Qualität der Ergebnisse eines Aggregationsalgorithmus, ergeben sich alternative Vorgehensweisen:

- Sind *alle* verfügbaren Kontexte stark verwandt, kann die Anwendung einen beliebigen Kontext auswählen und weiter verarbeiten. Eine Aggregation der typgleichen Kontexte kann vermieden werden, wenn aufgrund ihres hohen Verwandtschaftsgrades anzunehmen ist, dass eine Aggregation nicht sinnvoll ist. Auf diese Weise können potentielle Fehler durch die Aggregation abhängiger Kontexte verhindert werden; gleichzeitig kann Energie eingespart werden, da auf die Ausführung des Aggregationsalgorithmus verzichtet werden kann.
- Ist *ein Teil* der verfügbaren Kontexte stark verwandt, kann die Anwendung gezielt die Kontexte für eine Aggregation auswählen, welche die geringste Verwandtschaft aufweisen. So kann eine optimale Qualität bezüglich der Aggregationsfunktion erreicht werden.
- Liegt ein beliebiges Verwandtschaftsmuster vor, kann die Anwendung einen optimalen Algorithmus für die Weiterverarbeitung der verfügbaren Kontexte wählen.

Das zweite Grundmuster der informationellen Kontextabhängigkeit ist die *zyklische Verarbeitung*. Zyklische Verarbeitung von Kontexten tritt auf, wenn sich im Verarbeitungsgraph einer ubiquitären Informationsumgebung Kreise bilden, das heißt, wenn eine Anwendung einen Kontext konsumiert, an dessen Produktion sie bereits beteiligt war.

Abbildung 3.13 zeigt die schematische Darstellung der zyklischen Verarbeitung von Kontexten. Der Zyklus wird initiiert durch das Artefakt *D*, welches einen Kontext des Typs *c* bereitstellt. Dieser wird von Artefakt *A* konsumiert, das den Kontext *c* zu einem Kontext vom Typ *a* weiterverarbeitet. Kontext *a* wird wiederum von einem Artefakt vom Typ *B* konsumiert und ebenfalls weiterverarbeitet. Das Artefakt *B* produziert daraufhin einen Kontext des Typs *c*. Dieser Kontext kann nun wieder von Artefakt *A* konsumiert werden und schließt somit den Kreis.

Im Gegensatz zur Auffächerung und Multiplikation von Kontexten, lässt sich für die Kreisbildung kein typisches Anwendungsbeispiel angeben. Da im Allgemeinen in einer ubiquitären Informationsumgebung nicht bekannt ist, welche

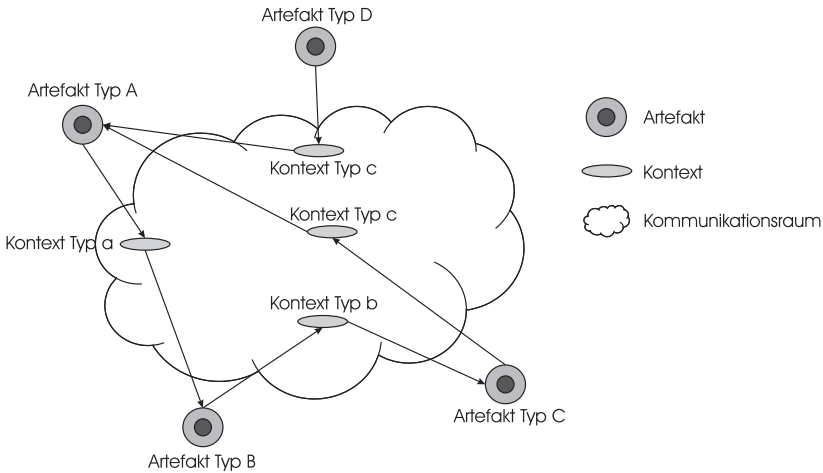


Abbildung 3.13. Zyklische Verarbeitung

Anwendungen verfügbar sind und welche Kontexte diese konsumieren und produzieren, ist auch die Detektion von Kreisen von nur zwei Artefakten nicht trivial. Lediglich Fälle von Artefakten, die Kontexte eines Typs konsumieren und gleichzeitig produzieren, lassen sich ohne Wissen über die Topologie des Verarbeitungsgraphen lösen. Diese Fälle können dann auftreten, wenn eine Anwendung einen Kontext nicht nur selbständig aus anderen Informationen ableiten kann, sondern, um zum Beispiel Energie für die Ableitung zu sparen, auch in der Lage ist, diesen Kontext, wenn er in der Umgebung vorhanden ist, direkt zu verwenden und ihn dann zu propagieren. Das Auftreten dieser direkten Schleifen aus nur einem beteiligten Artefakt kann leicht unterbunden werden, wenn die Anwendung Kontexte, die sie konsumiert und verarbeitet, ohne einen Kontext eines anderen Typs zu erzeugen, nicht propagiert.

Da Anwendungen ohne Hilfsmittel generell nicht in der Lage sind, die Bildung von Ringen in der Kontextverarbeitung zu erkennen, können die daraus entstehenden Abhängigkeiten nicht aufgelöst werden. Ist eine Anwendung dagegen in der Lage, den Grad der Verwandtschaft von Kontexten zu analysieren, kann sie einen Verarbeitungszyklus gezielt unterbrechen, wenn die Kontextabhängigkeiten

in diesem Zyklus zu einem nicht tolerierbaren Verlust an Kontextqualität führen würden.

Durch die Kombination der beiden vorgestellten Grundmuster der Kontextabhängigkeit in ubiquitären Informationsumgebungen lassen sich alle Situationen modellieren, in denen die Verwandtschaft von Kontextinformationen zu Problemen mit der inhaltlichen Kontextqualität bei der Ableitung führen kann. Es zeigt sich, dass die Verwandtschaft von Kontexten eine Eigenschaft der Kontextinstanz ist, die vom jeweiligen Ableitungsprozess bestimmt wird. Sie ist unabhängig von der semantischen Bedeutung der Kontextinformation und wird in dieser Arbeit als Kontextattribut – im Sinne der Definition 3.7 – eingeführt.

3.8.6 Zusammenfassung

In diesem Abschnitt wurde das, für diese Arbeit grundlegende Modell attributierter Kontexte eingeführt. Es wurden vier Kontextattribute identifiziert, die Eigenschaften von Kontexten in verschiedenen Dimensionen des Informationsraumes ubiquitärer Informationsumgebungen beschreiben. Diese Attribute sind:

- Alter
- räumliche Herkunft
- Verlässlichkeit
- Verwandtschaft

Sie zeichnen sich im Gegensatz zu anderen aus der Literatur bekannten Kontexteigenschaften dadurch aus, dass sie

- jedem Kontext, unabhängig von seiner Kontextklasse, zugeordnet werden können,
- die Eigenschaften einer individuellen Kontextinstanz beschreiben und
- unabhängig von der semantischen Bedeutung des Kontextes sind, den sie beschreiben.

Damit eignen sich die hier definierten Kontextattribute als Basis für die Entwicklung eines Kontextmanagementsystems, das unabhängig von semantischen Kontextmodellen, die Qualität von Kontexten in einer ubiquitären Informationsumgebung sichern kann.

3.9 Zusammenfassung

Dieses Kapitel stellt die theoretische Grundlage der vorliegenden Arbeit dar. Insbesondere wird hier das formale Modell von Kontext als Datenstruktur und das Modell attributierter Kontexte eingeführt.

Das hier vorgestellte funktionale Kontextmodell stellt Kontexte als Datenstruktur innerhalb eines informationstechnischen Systems dar. Es beschreibt die Kommunikation und Verarbeitung dieser Kontexte durch die Artefakte einer ubiquitären Informationsumgebung.

Ausgangspunkt für die Entwicklung dieses effizienten, funktionalen Kontextmodells war die Erstellung eines Kostenmodells der Kommunikation und Verarbeitung von Kontexten in ubiquitären Informationsumgebungen und die daraus resultierende Kostenrechnung für verschiedene Modellierungsansätze. Dabei geht die Modellierung auf die speziellen Anforderungen der Anwendungsdomäne des resultierenden Systems ein, indem zentrale Aspekte, wie die durch die Mobilität der Artefakte verursachte Dynamik in der Umgebung und die Skalierung, adressiert werden. Die wichtigsten Merkmale des funktionalen Modells sind:

- Entkoppelung der Kontextverarbeitung durch ein Produzent-Konsument-Schema.
- Entkoppelung der Kommunikation von Kontexten durch den CSpace.
- Repräsentation von Kontext als verarbeitbare Datenstruktur.

Aufbauend auf den Grundlagen des funktionalen Kontextmodells wurde das zentrale Modell attributierter Kontexte entwickelt. Attributierte Kontexte dienen als Basis für ein Qualitätssicherungssystem für Kontexte. Die Kontextattribute stellen allgemeine Eigenschaften von Kontexten dar, die von deren semantischer Bedeutung unabhängig sind. Dadurch kann die Kontextqualität unabhängig vom eingesetzten semantischen Kontextmodell evaluiert werden, wodurch eine hohe Flexibilität des Qualitätsmanagements erreicht werden kann.

Die definierten Kontextattribute repräsentieren verschiedene, für die Kontextverarbeitung relevante Dimensionen von Kontext, welche die Qualität von Kontexten beeinflussen können. Vier Kontextattribute wurden definiert:

- Alter
- räumliche Herkunft
- Verlässlichkeit
- Verwandtschaft

Neben der Unabhängigkeit von semantischen Modellen zeichnen sich diese Attribute dadurch aus, dass sie die Eigenschaften einer individuellen Kontextinstanz beschreiben und nicht einer ganzen Klasse von Kontexten, wie dies bei statistischen Maßen der Fall ist. Außerdem können die obigen Attribute jedem Kontext, unabhängig von seiner Kontextklasse, zugeordnet werden. Dies ermöglicht eine einheitliche Behandlung aller Kontexte auf Systemebene.

Im nun folgenden Kapitel dienen die hier entwickelten Modelle als Grundlage für die Umsetzung eines konkreten Qualitätsmanagementsystems für ubiquitäre Informationsumgebungen.

4 Architektur

In diesem Kapitel werden die vorgestellten Modelle zusammengeführt und in der Architektur einer ubiquitären Informationsumgebung mit integrierten Qualitätsmanagementmechanismen konkretisiert. Dabei werden die benötigten Verfahren und Algorithmen des Qualitätsmanagementsystems eingeführt und beschrieben, die dann als Grundlage für die Implementierung und Evaluierung des Gesamtsystems dienen.

Ziel der Architektur ist es, den Aufbau und die Abläufe innerhalb des Kontextmanagementsystems und der ubiquitären Informationsumgebung zu präzisieren. Das Kontextmanagementsystem wird dabei weitgehend in bestehende Architekturen ubiquitärer Informationssysteme eingebettet. Diese Architekturen werden wo erforderlich, erweitert und angepasst, um der Modellierung Rechnung zu tragen. Dies gilt insbesondere für die Anbindung des Qualitätsmanagementsystems an die semantische Kontexterkenkung, die sich auf die Spezifikation der nötigen Schnittstellen der Systeme beschränkt. Bei der Erarbeitung der Architektur des Kontextmanagementsystems mussten zahlreiche Entscheidungen hinsichtlich der Umsetzung der Modellierung getroffen werden, die Verfahren und Algorithmen zur Handhabung und Ausnutzung der Kontextattribute betrafen. Diese Entscheidungen wurden von den Anforderungen, welche von den speziellen Eigenschaften der Anwendungsdomäne gestellt werden, beeinflusst. Die Entscheidungsfindung beruht auf Ergebnissen von Versuchen aus den explorativen Arbeiten und der intensiven Simulation des Systems.

Die Beschreibung der Architektur gliedert sich in zwei Teile: Zuerst wird in kompakter Form die Architektur der zu Grunde liegenden ubiquitären Informationsumgebung anhand des Aufbau eines Artefaktes beschrieben. Dann werden das Modell des Kontextes als Datenstruktur mit den Modellen der Kontextverarbeitung und der entkoppelten Kommunikation verbunden und in Protokollen und Sprachen konkretisiert. Zentraler Bestandteil des Kontextmanagementsystems sind die Kontextattribute. Im zweiten Teil dieses Kapitels werden die einzelnen Attribute präzisiert und eine konkrete Umsetzung ihrer Funktion in der ubiquitären Informationsumgebung angegeben. Neben den Attributen „räumliche

Herkunft“ und „Kontextalter“, die in der Literatur bereits als interessante für die Beurteilung der Kontextqualität identifiziert wurden und hier funktional umgesetzt werden, führt diese Arbeit die „*Verlässlichkeit von Kontexten*“ und „*Verwandtschaft von Kontexten*“ als neue, zentrale Maße für die Bestimmung der Kontextqualität ein. Dabei werden die zur Qualitätsbestimmung und -überprüfung eingesetzten Algorithmen und Verfahren genau definiert und beschrieben. Sowohl Verlässlichkeit, als auch Verwandtschaft von Kontexten wird funktional definiert und algorithmisch umgesetzt so, dass konkrete quantitative Aussagen über diese Aspekte der Kontextqualität ermöglicht werden. Für die Bestimmung der Verlässlichkeit von Kontexten wird dabei auf Verfahren der unscharfen Logik zurückgegriffen, wogegen für die Bestimmung der Verwandtschaft von Kontexten Algorithmen entwickelt wurden, die Ideen aus dem Bereich der genetischen Algorithmen umsetzen. Dadurch wird die Funktion der Qualitätssicherung und ihre Schnittstellen zu der semantischen Kontextverarbeitung konkretisiert.

4.1 Architektur des Artefaktes und der ubiquitären Informationsumgebung

Dieser Abschnitt gibt einen Überblick über die Architektur einer ubiquitären Informationsumgebung anhand des Aufbaus der Artefakte, die diese Umgebung bilden. Andere aktive Komponenten, wie Infrastrukturobjekte, sind ähnlich aufgebaut. Ihnen fehlen zumeist nur einzelne Komponenten der Gesamtarchitektur wie sie von den Artefakten implementiert wird.

Abbildung 4.1 zeigt das Architekturdiagramm eines Artefaktes. Dabei liegt der grau hinterlegte Teil im zentralen Fokus dieser Arbeit. Da das hier entwickelte Kontextmanagementsystem Eigenschaften der Kommunikationsschicht nutzt, um seine Daten zu repräsentieren, muss diese auch näher betrachtet werden.

Die Kommunikationsschicht beherbergt die Schichten 1 bis 3 des ISO/OSI-Basisreferenzmodells [191, 1]. Da die Bitübertragungsschicht, die Sicherungsschicht und die Vermittlungsschicht hier nur am Rande betrachtet werden, werden sie zusammenfassend als **Kommunikationsschicht** bezeichnet.

Die **Kontextmanagementschicht** ist der zentrale Beitrag dieser Arbeit. Sie beinhaltet Verfahren, um auf der Basis der Kontextattribute die Qualität von Kontexten zu beurteilen und diese gegebenenfalls schon vor einer Weiterverarbeitung

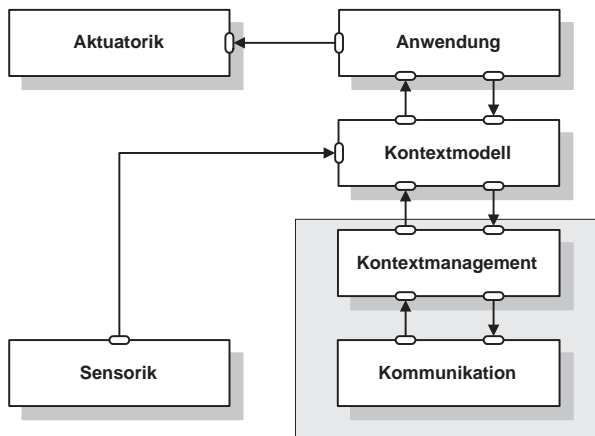


Abbildung 4.1. Architektur: Artefakt

durch die semantische Kontexterkennung und die Anwendung eines Artefaktes aus dem Verarbeitungsprozess herauszufiltern.

Die kontextverarbeitenden Schicht, die das semantische **Kontextmodell** beherbergt, ist für die Erkennung und Interpretation von Anwendungskontexten verantwortlich. Zusätzlich fließen in das Kontextmodell Sensordaten ein, die ein Artefakt selbst in der Umwelt sammeln kann. Das Kontextmanagementsystem stellt dem Kontextmodell Schnittstellen zur Verfügung, die den Zugriff auf die Qualitätsdaten eingehender Kontexte ermöglichen. Diese Qualitätsdaten versetzen die Kontextverarbeitung in die Lage, Anforderungen an die Kontextqualität zu formulieren und an das Qualitätsmanagementsystem zu kommunizieren.

Oberhalb der semantischen Kontextverarbeitung ist die **Anwendungsschicht** angesiedelt. Sie beinhaltet die funktionalen Komponenten eines Artefaktes, die auf Kontexte reagieren und an Kontextänderungen dynamisch angepasst werden. Die Anwendungsschicht steuert außerdem die Aktuatorik, die zur direkten Beeinflussung der Umwelt oder als Benutzerschnittstelle dient.

Die herausragende Eigenschaft der hier vorgestellten Architektur ist, dass neben der semantischen Kontextverarbeitung, welche Kontext für die Anwendungsschicht im Sinne kognitiver Kontexte interpretiert, sowohl das Kontextmanage-

ment, als auch die Kommunikationsschicht direkt auf relevante Aspekte der Kontexte zugreifen können und diese nutzen.

4.2 AwareCon und ConCom: Kontextkommunikation und Kontext-Raum

Die Grundlage für eine dezentral organisierte ubiquitäre Informationsumgebung bildet ein Kommunikationssystem, das die direkte Kommunikation (peer-to-peer) von Artefakten untereinander unterstützt. Ein solches System wurde mit AwareCon [17] speziell für die Erfordernisse des Ubiquitous Computing entwickelt.

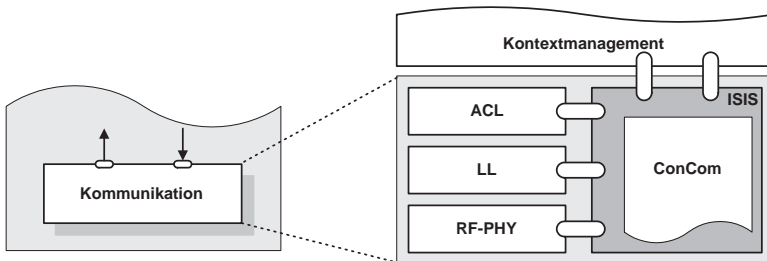


Abbildung 4.2. Architektur: AwareCon

Abbildung 4.2 zeigt die interne Architektur der Kommunikationsschicht. AwareCon implementiert die unteren Schichten des ISO/OSI Basisreferenzmodells. Zu den klassischen Netzwerkschichten kommt der ISIS (Internal Situation Store) hinzu, der die, über das Netzwerk kommunizierten Kontexte in Form von ConCom-Sätzen bereithält.

Bei ConCom [106] handelt es sich um eine Sprache zur Repräsentation und Kommunikation von Kontexten. Kontexte werden in ConCom tupelbasiert dargestellt, daher eignet sich die Sprache besonders gut zur Umsetzung des in Abschnitt 3.4.2 eingeführten Repräsentationsmodells von Kontexten. ConCom wurde im Rahmen dieser Arbeit um ein Tupelfeld zur Aufnahme der Kontextattribute erweitert. Aus den einzelnen Kontexttupeln C werden zur Kommunikation so genannte ConCom-Sätze gebildet. Sie dienen in Anlehnung an die natürliche

Sprache dazu, Situationen und Zusammenhänge von einzelnen Kontexten darzustellen.

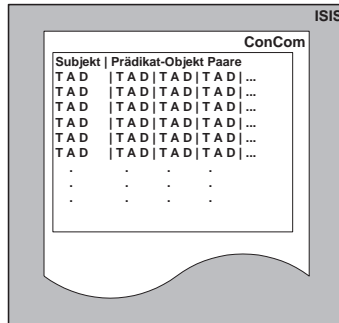


Abbildung 4.3. ConCom: Kontexttupel bilden ConCom-Sätze

Abbildung 4.3 zeigt ConCom-Sätze in der ISIS eines Artefaktes. Der ConCom-Satz beginnt dabei immer mit einem Subjekt, welches die Art des kommunizierenden Artefaktes näher beschreibt (Identitätskontext). Im Falle eines Boardmarkers aus dem AwareOffice würde das Subjekt übersetzt in natürliche Sprache „Ich bin ein Boardmarker“ bedeuten. Auf das Subjekt folgen beliebig viele Prädikat-Objekt-Paare, die aus je einem Kontext $C = \{T, A, D\}$ bestehen und das Subjekt näher beschreiben. Zum Beispiel: Subjekt: „Ich bin ein Boardmarker“; Prädikat-Objekt: „und mit mir wird gerade geschrieben!“.

AwareCon ermöglicht subscriptionsbasierte drahtlose Rundrufkommunikation. Die Subskription wird von einer höheren Schicht initiiert und über die SAPs der Kommunikationsschicht übergeben. AwareCon unterstützt die direkte Subskription auf das Typfeld T kommunizierter Kontexte. Die Kommunikation über AwareCon erfolgt grundsätzlich als Rundruf. Werden Daten auf dem Funkkanal empfangen, werden diese hinsichtlich der aktiven Subskriptionen ausgewertet. AwareCon gibt dann nur die Kontexte weiter an die Kontextmanagementschicht, für die eine Subskription vorliegen.

4.2.1 Realisierung des CSpace-Modells im Kommunikationskanal

Der Kontext-Raum – *CSpace* – wird in der hier vorgestellten Architektur direkt durch den Kommunikationskanal realisiert. Der Kommunikationskanal auf der Basis von AwareCon ist ein geteiltes Medium, auf das alle Teilnehmer desselben Netzwerksegmentes einer ubiquitären Informationsumgebung gleichberechtigten Zugriff haben.

Das Schreiben in den CSpace, das durch den Operator `cspace.add(C)` ausgedrückt wird, erfolgt durch das Senden eines Kontexttupels *C* durch eines der Artefakte. Der gesendete Kontext ist dann im CSpace verfügbar und kann von allen Artefakten der ubiquitären Informationsumgebung empfangen werden.

Grundlage für den Empfang eines Kontextes ist im CSpace die Subskription mittels des Operators `cspace.subscribe(t)`. Dieser entspricht dem, durch AwareCon bereitgestellten, Subskriptionsmechanismus auf Kontexttypen. Ein Kontext im CSpace beziehungsweise im Kommunikationskanal der ubiquitären Informationsumgebung kann somit von allen Artefakten empfangen werden, die seinen Kontexttyp *t* abonniert haben. Individuelle Kontexte stehen so für die Dauer eines AwareCon RF-Übertragungszeitschlitzes (RF-Slot) [17] im CSpace bereit.

Das Aufrufen des, im Modell für die Servicefunktionen, zuständigen Operators `cspace.maintain()` geschieht implizit mit Ende des RF-Slots. Nach dem Ablauf dieser Zeitspanne, ist der im CSpace befindliche Kontext nicht mehr verfügbar. Der Kommunikationskanal auf der Basis von AwareCon bildet somit einen CSpace mit einer Kanalzeit von `channel_time = 1` RF-Slot, der die entkoppelte Kommunikation der Artefakte einer ubiquitären Informationsumgebung unterstützt.

Zusammen mit dem, durch AwareCon realisierten subskriptionsbasierten Zugriff auf die kommunizierten Kontexte, erfüllt die hier vorgestellte Architektur der Kommunikationsschicht die Anforderungen an die Kontextkommunikation in ubiquitären Informationsumgebungen, welche in Abschnitt 3.3.1 und im Kostenmodell im Abschnitt 3.2.2 formuliert wurden.

4.3 Architektur des Kontextmanagementsystems

Die zentrale Komponente dieser Arbeit, welche die Algorithmen und Verfahren zur Sicherung der Kontextqualität in ubiquitären Informationsumgebungen beinhaltet, ist das Kontextmanagementsystem. In der Gesamtarchitektur fügt sich die Kontextmanagementschicht zwischen der Kommunikationsschicht, welche den netzwerkseitigen Transport und die Subskription auf Kontexttypen organisiert und der Schicht der semantischen Kontextverarbeitung, ein.

Abbildung 4.4 zeigt die Architektur dieser Schicht. Über den Zugangspunkt von der Kommunikationsschicht zur Kontextmanagementschicht, erhält das Kontextmanagementsystem Kontexte der Form $C = \{T, A, D\}$, welche das Artefakt zuvor abonniert hat. Nachdem die Kommunikationsschicht das Feld T des Kontextes zur Filterung der abonnierten Kontexte ausgewertet hat, arbeitet das Kontextmanagementsystem auf dem Feld A , welches die Kontextattribute enthält.

In Kapitel 3 wurden vier Attribute als Dimensionen von Kontexten modelliert, die der Qualitätssicherung dienen können:

- Temporale Dimension
- Räumliche Dimension
- Verlässlichkeit von Kontexten
- Abhängigkeit von Kontexten

Aus diesen werden nun für die Systemarchitektur konkrete Attribute abgeleitet, welche die im Modell beschriebenen Eigenschaften der einzelnen Dimensionen von Kontext repräsentieren und für ein Qualitätsmanagementsystem nutzbar machen. Für die Beschreibung der Architektur der Kontextmanagementschicht werden hier die Attribute nur kurz benannt und ein Überblick über ihre Funktion gegeben. Die ausführliche Diskussion der einzelnen Attribute und der mit ihnen verknüpften Verfahren und Algorithmen folgt ab Abschnitt 4.4.

Die funktionalen Komponenten, in denen die Verarbeitung der einzelnen Kontextattribute geschieht, sind in der Abbildung als Rauten dargestellt. Die temporale Dimension von Kontexten wird durch das Attribut *Kontextalter* (engl. *Context Age*, CA) repräsentiert. Das Attribut Kontextalter wird von der, in Abbildung 4.4 mit CA bezeichneten, funktionalen Komponente verarbeitet.

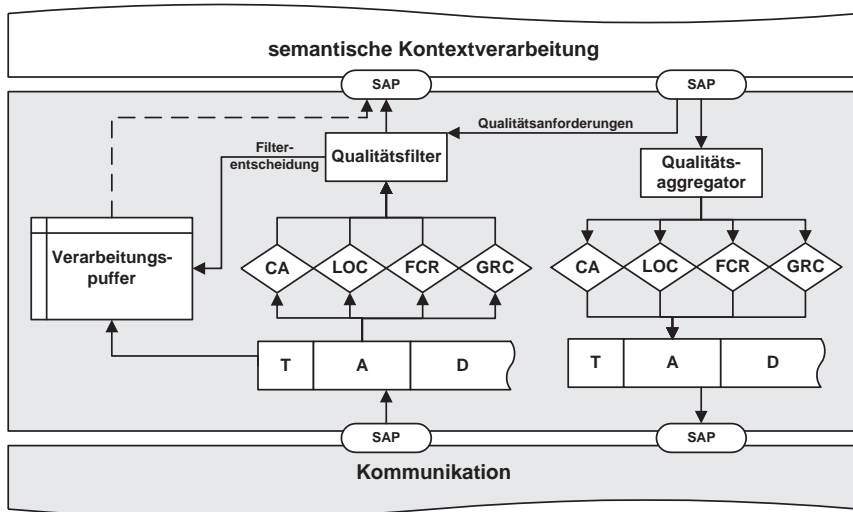


Abbildung 4.4. Architektur: Kontextmanagementschicht

Die Komponente LOC verarbeitet das Attribut *räumliche Herkunft* (engl. *Location, LOC*), welches die Umsetzung der räumlichen Dimension von Kontexten darstellt.

Die Verlässlichkeit von Kontexten wird durch ein Attribut mit dem Namen *Fuzzy Context Reliability (FCR)* repräsentiert. Das zu diesem Attribut gehörende Modul verarbeitet Verlässlichkeitsmaße auf der Basis von Verfahren aus der unscharfen Logik (engl. *Fuzzy Logic*).

Als viertes Kontextattribut wird die *genetische Verwandtschaft von Kontexten* (engl. *Genetic Relation of Contexts, GRC*) eingeführt. Dieses Attribut adressiert die Abhängigkeit von Kontexten. Das Modul GRC in Abbildung 4.4 implementiert Verfahren, die an Ideen aus der Genetik und aus dem Bereich der genetischen Algorithmen angelehnt sind.

4.3.1 Empfang eines Kontextes

Nach der Annahme eines Kontextes durch die Kontextmanagementschicht, werden die Attributwerte a aus dem Attributfeld A des Kontextes $C = \{T, A, D\}$ gelesen und an die entsprechenden funktionalen Module (CA, LOC, FCR und GRC) weitergeleitet. Die funktionalen Module werten dann die Attributwerte aus, während der Kontext selbst in einem Verarbeitungspuffer zwischengespeichert wird. Die Ergebnisse der Auswertung der Kontextattribute werden im Qualitätsfilter aggregiert.

Die Aufgabe des Qualitätsfilters ist es, Kontexte, welche nicht den Qualitätsanforderungen der Anwendung genügen, auszufiltern. Zu diesem Zweck kann der Qualitätsfilter über den SAP von der darüber liegenden Schicht konfiguriert werden. Die Qualitätsanforderungen werden dabei in Form von Schwellwerten für die verschiedenen Attributwerte an den Qualitätsfilter übergeben. Es können sowohl obere als auch untere Grenzen, sowie Wertebereiche konfiguriert werden. Ebenso ist die logische Verknüpfung von Anforderungen an die einzelnen Attribute möglich.

Erkennt der Qualitätsfilter, dass ein Kontext nicht die Anforderungen der Anwendung erfüllt, verwirft er ihn, indem er den Kontext aus dem Verarbeitungspuffer löscht. Erfüllt der bearbeitete Kontext die Anforderungen, wird er an die Schicht zur semantische Kontextverarbeitung über den entsprechenden SAP weitergegeben. Dabei wird der vollständige Kontext C , inklusive des Attributfeldes A ,

so übergeben, wie er von der Kommunikationsschicht entgegengenommen wurde. Zusätzlich stellt der Qualitätsfilter die Ergebnisse der Verarbeitung der Attributwerte gesondert bereit, damit diese als zusätzliche kontextuelle Information in die semantische Verarbeitung eingehen können.

4.3.2 Versand eines Kontextes

Wird ein Kontext durch ein Artefakt nach außen kommuniziert, muss dieser mit den benötigten Qualitätsinformationen ausgestattet werden. Dies geschieht, indem die Werte der Kontextattribute durch das Kontextmanagement bestimmt und entsprechend gesetzt werden.

Hierzu werden alle Informationen zur Berechnung der Attributwerte und der zu kommunizierende Ausgangskontext von der semantischen Kontextverarbeitung an den Qualitätsaggregator der Kontextmanagementschicht übergeben. Die Daten, welche vom Qualitätsaggregator benötigt werden, umfassen neben dem Ausgangskontext die Eingangskontexte, aus denen dieser abgeleitet wurde¹ und gegebenenfalls Konfigurationsparameter für die einzelnen Module zur Berechnung der Attributwerte. Die Konfigurationsmöglichkeiten, die hier zur Verfügung stehen, werden eingehend bei der Beschreibung der einzelnen Attribute diskutiert.

Wie Abbildung 4.4 zeigt, werden die Daten dann funktionalen Modulen (CA, LOC, FCR und GRC) zugeführt, welche die konkrete Berechnung der Attributwerte des Ausgangskontextes vornehmen. Sind alle Attributwerte berechnet, werden diese im Feld A des Ausgangskontextes gesetzt. Dann wird der Kontext am unteren SAP der Kommunikationsschicht zur Übertragung übergeben.

4.4 Aufbau der Kontextattribute

In den folgenden Abschnitten wird der konkrete Aufbau der einzelnen Kontextattribute beschrieben. Dabei werden die Algorithmen und Verfahren eingeführt, die im Laufe dieser Arbeit entwickelt wurden, um auf der Basis der Attribute ein Qualitätsmanagementsystem für Kontexte in ubiquitären Informationsumgebungen zu realisieren.

¹Die Eingangskontexte können zu diesem Zweck auch direkt dem Verarbeitungspuffer der Kontextmanagementschicht entnommen werden.

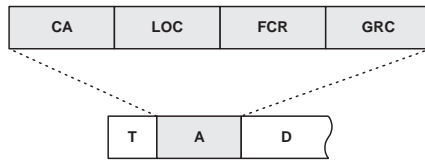


Abbildung 4.5. Kontextattribute im Feld A eines Kontextes *C*

Ziel des Qualitätsmanagementsystems ist es, die Kontexterkennungsrate von Anwendungen in ubiquitären Informationsumgebungen zu verbessern. Dazu werden von der Anwendung Qualitätsanforderungen festgelegt, die innerhalb der Kontextmanagementschicht durch den Qualitätsfilter überprüft werden. Sind die Anforderungen einer Anwendung an die Kontextqualität nicht erfüllt, wird der betreffende Kontext verworfen. Er geht somit nicht mehr in die semantische Kontextverarbeitung des Artefaktes ein.

Die in Abschnitt 3.8 eingeführten Kontextattribute, dienen zur Bestimmung und Repräsentation der Kontextqualität. Ihre Auswahl und Modellierung trägt relevanten Problemen in realen ubiquitären Informationsumgebungen Rechnung, welche die Qualität von Kontexten beeinträchtigen können. Dem Aufbau des hier vorgestellten Systems liegt die Definition 2.2 der Kontextqualität aus Abschnitt 2.6.1 zu Grunde.

Wichtig ist hierbei, dass die Qualität eines Kontextes grundsätzlich durch den Kontextkonsumenten bestimmt wird. Der Produzent stattet einen Kontext, den er in einer ubiquitären Informationsumgebung bereitstellt, mit den Daten aus, die es den Konsumenten erlauben, ein für sie gültiges Qualitätsmaß zu errechnen und dieses zu bewerten.

Diese wichtige architektonischen Entscheidung beruht auf der Beobachtung, dass die Beurteilung der Qualität von Kontexten signifikant von den individuellen Anforderungen der verarbeitenden Anwendung abhängt und weniger von Faktoren, die der Produzent eines Kontextes objektiv beurteilen kann. Daher beschränkt sich das hier entwickelte Qualitätsmanagementsystem auf Seiten des Kontextproduzenten auf eine geeignete Vorverarbeitung der Attributwerte, die den Konsumenten die Qualitätsbeurteilung erleichtert.

Ein weiterer wichtiger Punkt, der den Entwurf der im Folgenden vorgestellten Verfahren beeinflusst hat, sind die Rahmenbedingungen, welche von der adres-

sierten Anwendungsdomäne (vgl. Abschnitt 2.1.4) vorgegeben werden. Vor allem wurde versucht der Skalierung und den Einschränkungen durch die begrenzten Ressourcen der Artefakte Rechnung zu tragen. Das Ergebnis sind Algorithmen und Verfahren, die bei geringer Ressourcenbelastung in der Lage sind, auch in hoch skalierten ubiquitären Informationsumgebungen effiziente Qualitätsmaße zu generieren, deren Ausnutzung zu einer Verbesserung der Kontexterkenntnisleistung von Artefakten führt.

Im Folgenden werden nun die vier Kontextattribute eingeführt. Dabei wird die Funktion der Eingabe- und Ausgabemodule eines Attributes aus Abbildung 4.4 in jeweils einem Abschnitt zusammengefasst, um das Zusammenspiel der Erzeugungs- und Auswertungsalgorithmen besser darzustellen.

4.5 Kontextalter (CA)

Durch das Attribut Kontextalter wird die temporale Dimension von Kontext repräsentiert und für das Qualitätsmanagementsystem zugänglich gemacht. Wie bereits in der Modellbildung in Abschnitt 3.8.2 erläutert wurde, wird die temporale Einordnung und Bewertung von Kontexten von zwei wichtigen Größen beeinflusst: dem Alter des Kontextes und seiner Ableitungsdauer.

Um auf Seiten der Kontextkonsumenten die Berechnung beider Größen möglichst effizient zu gestalten, enthält das Attribut Kontextalter zwei Zeitstempel (siehe Abbildung 4.6): das Keimalter (engl. Context Seed Age (CSA)) und das Produktionsalter (engl. Context Production Age (CPA)).

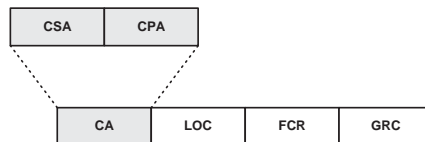


Abbildung 4.6. Attribut Kontextalter (CA) mit Zeitstempeln: Context Seed Age (CSA) und Context Production Age (CPA)

Das CSA eines Kontextes C enthält den Zeitpunkt, zu dem der älteste Kontext C_0 produziert wurde, der Teil des Ableitungsbaumes von C ist. CPA ist der Zeitpunkt, zu dem der Kontext C selbst produziert wurde. Zu beachten ist, dass es

sich hier trotz der Bezeichnung „Alter“ um Zeitstempel handelt. Abbildung 4.7 zeigt, wie das CSA und das CPA eines Kontextes höherer Ordnung C , zeitlich angeordnet sind.

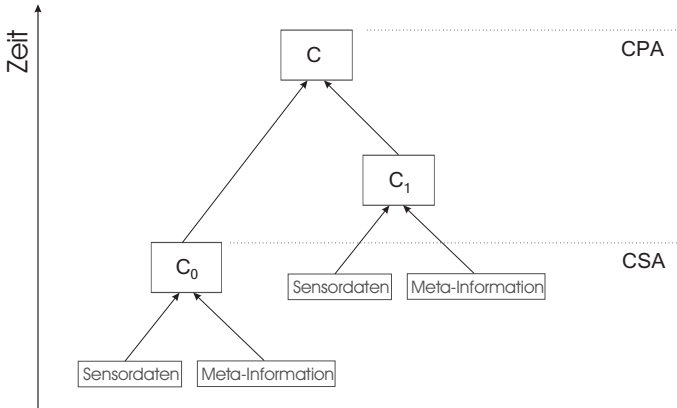


Abbildung 4.7. Context Seed Age (CSA) und Context Production Age (CPA) eines Kontextes C

4.5.1 Setzen des Kontextalters

Auf welchen konkreten Wert CSA bei der Produktion eines Kontextes gesetzt wird, hängt davon ab, ob es sich um einen Kontext erster Ordnung handelt, der nur aus Sensordaten und Metainformationen abgeleitet wird, oder ob ein Kontext höherer Ordnung produziert wird, der von anderen Kontexten abgeleitet wird. Das CPA wird in jedem Fall, unabhängig von seiner Ordnung, auf den Zeitpunkt der Produktion des Kontextes gesetzt.

Wird ein Kontext erster Ordnung C_0 produziert, werden, wie in Abbildung 4.8 zu sehen ist, sowohl das CSA, als auch das CPA auf den Zeitpunkt seiner Produktion gesetzt. Der Qualitätsaggregator stellt dazu die aktuelle Zeit der Echtzeituhr des produzierenden Artefaktes t bereit, die dann durch das Modul CA in die Felder $CSA(C_0)$ und $CPA(C_0)$ des Kontextes C_0 kopiert wird.

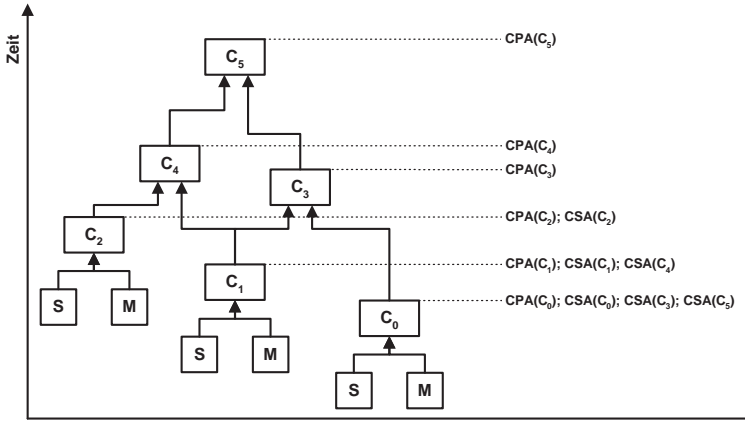


Abbildung 4.8. Kontextalter verschiedener Kontexte eines Ableitungsbaumes

Wird ein Kontext, zum Beispiel C_5 , höherer Ordnung produziert, wird, wie oben beschrieben, $CPA(C_5)$ auf die aktuelle Zeit gesetzt. CSA wird allgemein nach der folgenden Formel berechnet:

$$CSA(C) = \min\{(CSA(C_0), CSA(C_1), CSA(C_2), \dots, CSA(C_n))\} \quad (4.1)$$

Wobei $C_0, C_1, C_2 \dots C_n$ die Kontexte bezeichnet, die ein Artefakt konsumiert hat, um aus ihnen den Kontext C abzuleiten. Bezogen auf das Beispiel in Abbildung 4.8 führt dies zu folgenden Werten für die CSA :

$$CSA(C_5) = \min\{(CSA(C_0), CSA(C_1), CSA(C_2), CSA(C_3), CSA(C_4))\} = CSA(C_0)$$

$$CSA(C_4) = \min\{(CSA(C_1), CSA(C_2))\} = CSA(C_1)$$

$$CSA(C_3) = \min\{(CSA(C_0), CSA(C_1))\} = CSA(C_0)$$

$$CSA(C_2) = CAP(C_2)$$

$$CSA(C_1) = CAP(C_1) \quad \text{und}$$

$$CSA(C_0) = CAP(C_0)$$

4.5.2 Auswertung des Kontextalters

Bei der Auswertung des Kontextattributes Kontextalter, werden im Modul CA das Alter eines konsumierten Kontextes und seine Ableitungsdauer (engl. Total Context Derive Time ($TCDT$)) aus CSA und CPA berechnet und an den Qualitätsfilter weitergeleitet.

Das Kontextalter berechnet sich aus der Differenz der aktuellen Zeit t und CPA . Es gibt an, wie lange der Kontext zwischen seiner Produktion und seiner Verarbeitung schon in der ubiquitären Informationsumgebung verfügbar war. CSA und CPA erlauben es, die Zeit zu berechnen, die nötig war, um einen Kontext abzuleiten, die $TCDT$. Dabei gilt:

$$TCDT(C) = CPA(C) - CSA(C) \quad (4.2)$$

Die Berechnung der $TCDT$ erfolgt durch den Konsumenten eines Kontextes aus CSA und CPA . Alternativ könnte die $TCDT$ direkt als CA-Feld jedem Kontext beigelegt werden. Gegen diesen Ansatz spricht allerdings der erhöhte Rechenaufwand: Wären nur CPA und $TCDT$ als CA-Felder verfügbar, müsste der Produzent des Kontextes die $TCDT$ neu berechnen, indem er

$$TCDT(C_{neu}) = \max\{(t - CPA(C_{alt0}) + TCDT_{alt0}), (t - CPA(C_{alt1}) + TCDT_{alt1}), \dots, (t - CPA(C_{altn}) + TCDT_{altn})\}$$

bildet.

Dabei muss für jeden Quellkontext, der in C_{neu} eingeht $t - CPA(C_{alt}) + TCDT_{alt}$ berechnet und über diese Werte das Maximum gebildet werden. Bei n Quellkontexten führt dies zu einem Rechenaufwand von n Additionen und n Subtraktionen für die Berechnung der $(t - CPA(C_{altn}) + TCDT_{altn})$. Dazu kommen $\frac{(n-1)n}{2}$ Vergleiche für die Berechnung des Maximums. Dem gegenüber stehen bei der hier vorgeschlagenen Methode ebenfalls $\frac{(n-1)n}{2}$ Vergleiche für die Berechnung des Minimums der CSA , dafür aber lediglich eine einzige Subtraktion für die Berechnung von $TCDT$. Damit würde eine Speicherung von $TCDT$ bei jeder Kontextableitung in einer ubiquitären Informationsumgebung zu einem Mehraufwand von $2n - 1$ Additionen und Subtraktionen gegenüber der Speicherung von CSA führen.

Nach der Berechnung leitet das Modul CA das Kontextalter und die Ableitungsdauer an den Qualitätsfilter weiter.

4.6 Räumliche Herkunft (LOC)

Das Attribut räumliche Herkunft (siehe Abbildung 4.9) erlaubt dem Qualitätsmanagementsystem den Zugriff auf Informationen, welche die räumliche Dimension von Kontexten betreffen. Wie in der Modellierung der räumlichen Dimension von Kontext Abschnitt 3.8.3 gezeigt wurde, organisiert der Mensch seine Umgebung in räumlichen Begriffen. Daher haben räumliche Zusammenhänge eine wichtige Bedeutung bei der Erfassung und Wiedergabe von Situationen. Dabei spielen Begriffe, die zur räumlichen Organisation dienen, wie Abstand und Nähe, eine wichtige Rolle.

CA	LOC	FCR	GRC
----	-----	-----	-----

Abbildung 4.9. Attribut räumliche Herkunft (LOC)

4.6.1 Lokationsmodell des RAUM-Systems

In dem hier vorgestellten System wird die Herkunft von Kontexten mit Hilfe des RAUM-Systems dargestellt [13, 14, 184]. Das dort verwendete Lokationsmodell erlaubt es, Orte und Bereiche, sowie deren Beziehungen sowohl auf semantischer, als auch auf geometrischer Ebene, darzustellen. Das RAUM-System dient ursprünglich zur lokationsbasierten Kommunikation in interaktiven Räumen [14]. Es eignet sich aber auch besonders gut zur Erweiterung anderer Kontextkommunikationssysteme um Funktionen, wie der lokationsbasierten Adressierung und der Bereicheskommunikation [184, 18].

Lokationen, also Punkte in der räumlichen Umgebung, werden im RAUM-Lokationssystem mit Hilfe eines Lokationsbaumes, wie ihn Abbildung 4.10 zeigt, dargestellt [18]. Der Lokationsbaum besteht aus drei Schichten: einer Wurzel, hierarchisch geordneten Ebenen, die semantische Lokationsangaben enthalten und einer Ebene geometrischer Angaben, durch welche Lokationen verfeinert werden können. Bis zu drei semantische Ebenen und die Positionsangabe in kartesischen Koordinaten sind optional, um eine möglichst große Flexibilität des Systems zu erreichen.

Formal wird der Lokationsbaum von einer Inklusionsrelation Ξ induziert: Seien o, p, q Lokationen und O, P, Q Mengen von Lokationen $O := \{a, b, \dots\}$, so

genannte Gebiete. Sei weiter \mathfrak{P} die Potenzmenge dieser Gebiete. Dann ist Ξ definiert als eine Relation auf \mathfrak{P} mit:

1. $\forall O \in \mathfrak{P} : \langle O, O \rangle \notin \Xi$ (irreflexiv)
2. $\forall O, P, Q \in \mathfrak{P} : \langle P, Q \rangle \in \Xi \wedge \langle O, P \rangle \in \Xi \Rightarrow \langle O, Q \rangle \in \Xi$ (transitiv)
3. $\forall O, P \in \mathfrak{P} : \langle P, O \rangle \in \Xi \Rightarrow \langle O, P \rangle \notin \Xi$ (asymmetrisch)
4. $\forall O, P, Q \in \mathfrak{P} : \langle P, O \rangle \in \Xi \wedge \langle Q, P \rangle \in \Xi \Rightarrow P \cap Q = \emptyset$ (disjunkt)
5. $\forall O, P, Q \in \mathfrak{P} : \langle P, O \rangle \in \Xi \wedge \langle Q, P \rangle \in \Xi \wedge \exists X \in \Xi : \langle X, P \rangle \in \Xi \Rightarrow \langle X, Q \rangle \notin \Xi$

Ξ bildet eine hierarchische Mengenrelation auf \mathfrak{P} wobei 1. bis 3. eine strikte partielle Ordnung auf der Potenzmenge der Gebiete definieren. 4. schränkt den durch 1. bis 3. entstehenden Verband von Gebieten ein, indem nur die Ableitung von disjunkten Gebieten erlaubt wird. 5. verbietet die Ableitung von mehr als einem Elternknoten (Mehrfachvererbung). Die resultierende Relation Ξ kann als Baumstruktur visualisiert werden, in der die rechte Seite des Ausdrucks $\langle P, O \rangle$ eine Ebene höher notiert wird, als die linke Seite. Eine noch ausführlichere Beschreibung des RAUM-Lokationsmodells und seiner Funktionen findet sich in [18].

Die Wurzel des Lokationsbaumes beschreibt den semantischen Bereich, den der Baum abdeckt und für den er Lokationsangaben bereitstellen kann. Die Lokationsangaben eines bestimmten Artefaktes werden im RAUM-System durch einen Pfad im Lokationsbaum repräsentiert. Ein solcher Pfad beginnt an der Wurzel des Baumes und endet an einem beliebigen Knoten oder einem Blatt. Ein Beispiel für einen solchen Pfad ist in Abbildung 4.10 grau hinterlegt dargestellt. Je länger der Pfad ist, der eine Lokation angibt, desto genauer ist diese spezifiziert.

4.6.2 Setzen der räumlichen Herkunft

Das Attribut räumliche Herkunft wird in einem 32-44 Byte langen Feld gespeichert, das wie in Abbildung 4.11 dargestellt, aufgebaut ist. Die Identifikation der Baumwurzel (Root-ID) kann je nach Implementierung 4-16 Bytes umfassen. Durch die Root-ID können ganze ubiquitäre Informationsumgebungen identifiziert werden, die durch einen Lokationsbaum repräsentiert werden. In diesem

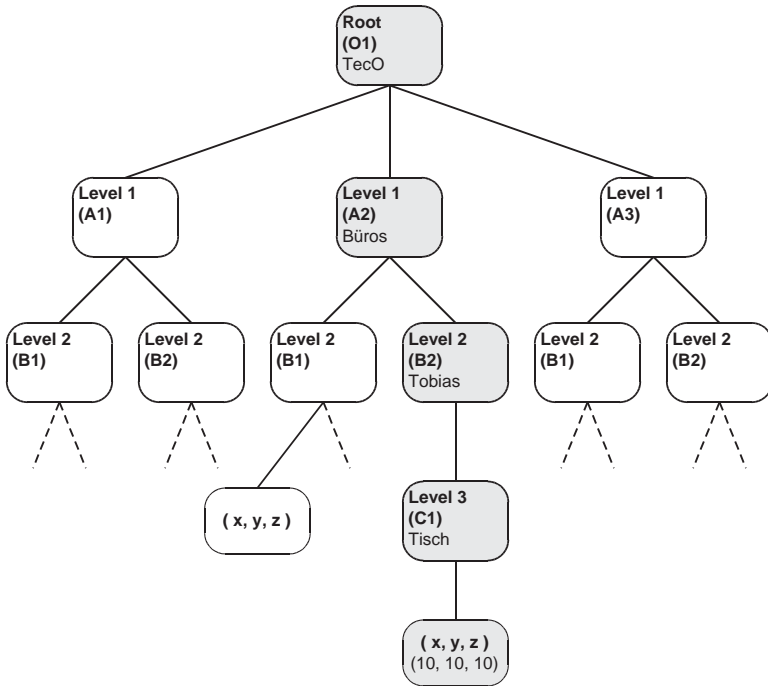


Abbildung 4.10. Beispiel eines Lokationsbaumes

Fall kann die Root-ID auch als Kommunikationsadresse für den Datenverkehr zwischen verschiedenen ubiquitären Informationsumgebungen eingesetzt werden [184].

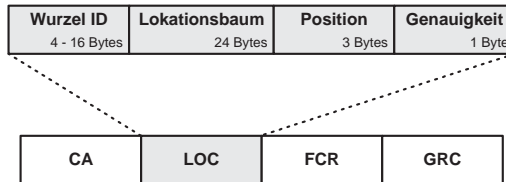


Abbildung 4.11. Aufbau der RAUM-Lokationsbeschreibung als Kontextattribut

Die Root-ID wird gefolgt vom Feld „Lokationsbaum“. In diesem Feld ist ein Pfad durch den semantischen Teil eines Lokationsbaumes einer ubiquitären Informationsumgebung gespeichert. Diese Beschreibung wird komprimiert in 3-2-Codierung abgelegt (vgl. [14]). Die 3-2-Codierung erlaubt es, 3 Zeichen eines reduzierten Zeichensatzes in 2 Byte zu speichern. Das Feld „Position“ enthält kartesische Koordinaten, durch welche die Position des Artefaktes genauer spezifiziert werden kann. Die „Genauigkeit“ gibt die Auflösung des, der Lokationsangabe zu Grunde liegenden, Lokationssystems an.

Das RAUM-Lokationsmodell ist, unabhängig von dem verwendeten Lokationssystem zur Positionsbestimmung, einsetzbar. Wird ein neuer Kontext erzeugt, fordert der Qualitätsaggregator die aktuelle Position des Artefaktes an und übergibt diese an das Modul LOC. Dieses übersetzt, falls nötig, die Lokationsangabe in ein RAUM-konformes Format und speichert diese im Attributfeld LOC des Kontextes.

4.6.3 Auswertung der räumlichen Herkunft

Zur Auswertung des Attributes räumliche Herkunft, extrahiert das Modul LOC die Lokationsbeschreibung im RAUM-Format und gibt diese an den Qualitätsfilter weiter. Der Qualitätsfilter übernimmt dann die Auswertung der Lokationsinformation anhand der Qualitätsvorgaben der Anwendung. Dazu definiert die Anwendung ihre Qualitätsvorgaben als RAUM-konforme Interessensbereiche. Diese legen fest, aus welchen räumlichen Gebieten Kontexte akzeptiert werden können.

Interessensbereiche im RAUM-System wurden ursprünglich entwickelt, um Daten in RAUM-basierten, lokationsadressierten Kommunikationssystemen routen zu können [184].

Einer der Vorteile des RAUM-Systems ist, dass Anforderungen wie: „Akzeptable Kontexte müssen im selben Raum produziert worden sein, in dem sich das konsumierende Artefakt befindet!“ direkt als semantischer Interessensbereich abgebildet werden können. Die so formulierte Qualitätsanforderung bleibt stabil, auch wenn das Artefakt mobil ist und selbst zwischen verschiedenen Räumen wechselt. Der Qualitätsfilter prüft, ob ein zu konsumierender Kontext aus den so definierten Interessensbereichen stammt. Ist dies nicht der Fall, wird der Kontext verworfen. Ansonsten wird er an die semantische Kontextverarbeitung weitergegeben.

4.7 Verlässlichkeit von Kontexten (FCR)

Im Gegensatz zu den Kontextattributen Kontextalter (CA) und räumliche Herkunft (LOC) konnte bei der Entwicklung der Algorithmen und Verfahren für die Bereitstellung eines Verlässlichkeitsmaßes für Kontexte nicht auf bereits bekannte Ansätze zurückgegriffen werden. Ziel des Attributes Verlässlichkeit von Kontexten (FCR) (Abbildung 4.12) ist es, ein Verlässlichkeitsmaß für individuelle Kontextinstanzen im Sinne der Definition 3.16 bereitzustellen. In diesem Abschnitt wird ein solches Maß in Anlehnung an [20] entwickelt.

CA	LOC	FCR	GRC
----	-----	-----	-----

Abbildung 4.12. Attribut Verlässlichkeit von Kontexten (FCR)

4.7.1 Verlässlichkeit in unscharfer Logik

Die Verlässlichkeit von Kontexten wird mit Hilfe der unscharfen Logik (Fuzzy Logik) [182] modelliert. Der Vorteil gegenüber der Verwendung scharfer Werte zur Beschreibung der Verlässlichkeit von Kontexten liegt in der natürlichen Abbildung der Eigenschaften von Kontextübergängen.

In Abschnitt 3.8.4 wurde Verlässlichkeit als Dimension von Kontext modelliert und das Verlaufsmodell der Kontextverlässlichkeit vorgestellt. Dieses Modell hat schematisch gezeigt, dass der Übergang zwischen verschiedenen Kontexten der realen Welt nur bis zu einer bestimmten Grenzauflösung beschrieben werden kann. Diese Grenzauflösung der für die semantische Kontexterkennung verwendeten Algorithmen wird durch die Anwendung und die Art des Artefaktes, welches sie implementiert, vorgegeben.

Das hier eingeführte algorithmische Verfahren zur Bestimmung eines quantitativen Verlässlichkeitsmaßes für Kontextinstanzen benutzt ein auf Fuzzy Logik basierendes Expertensystem, um die Ausgabe eines beliebigen Kontextalgorithmus in Bezug auf die Eingabe zu bewerten. Wie bereits in Abschnitt 2.5.1 gezeigt wurde, führen Kontextalgorithmen eine Klassifikation ihrer Eingabedaten durch. Das Ergebnis ist die Zuordnung der Eingabe zu einer bestimmten Kontextklasse, deren semantische Bedeutung in semantischen Kontextmodellen verwaltet wird. Angesichts der Eigenschaften des Verlaufsmodells der Kontextverlässlichkeit (Abschnitt 3.8.4), liegt es nahe, den Abstand des Eingabevektors eines Kontextalgorithmus zum Zentrum der zugeordneten Klasse, als Maß für die Verlässlichkeit der Erkennung dieser Kontextinstanz zu definieren.

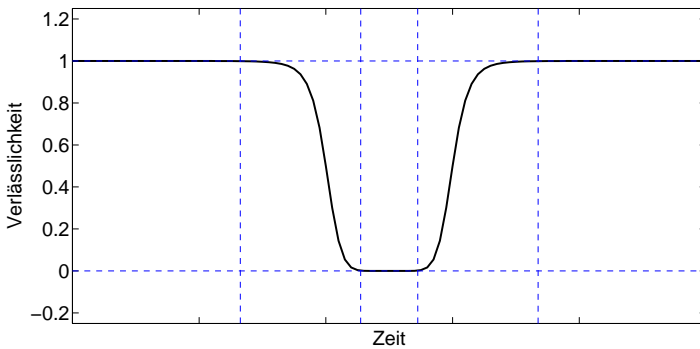


Abbildung 4.13. Verlauf der Verlässlichkeit von Kontext

Abbildung 4.13 zeigt noch einmal den Verlauf der so definierten Verlässlichkeit unter Annahme eines optimalen Kontexterkenners. Dabei ist der Verlässlichkeit

ein Wert von 1 zugeordnet, wenn der erkannte Kontext und der reale Kontext übereinstimmen. Ein Verlässlichkeitswert von 0 zeigt dagegen an, dass der Abstand des Eingavektors des FCR-Systems maximale Entfernung zu den umliegenden Klassenzentren der Kontexterkennung hat.

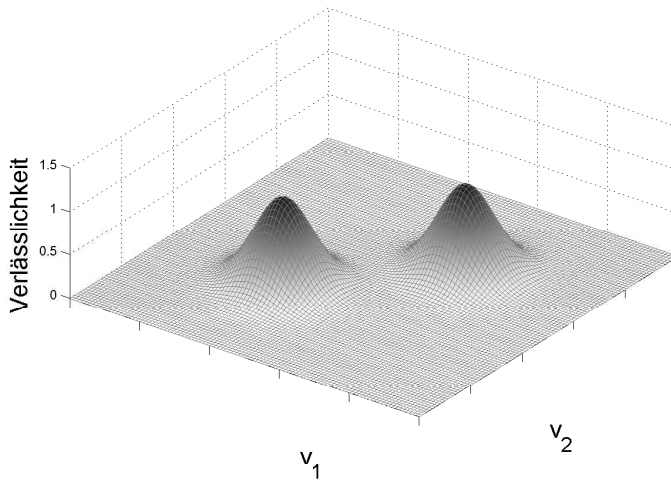


Abbildung 4.14. Verlässlichkeit von Kontextclustern

Die Bestimmung dieses Abstandes geschieht hier, unabhängig vom Kontextalgorithmus, mittels eines Fuzzy Expertensystems (Fuzzy Inference System, FIS). Die Verlässlichkeit einer Kontexterkennung kann dabei durch den Grad der Zugehörigkeit des Kontextes zu einer Kontextklasse ausgedrückt werden. Abbildung 4.14 zeigt einen Informationsraum der Dimension 2 (v_1 und v_2) mit 2 darin eingebetteten Kontextklassen. Die Verlässlichkeit von Eingavektoren \vec{v} bezüglich dieser Klassifizierung ist auf der z-Achse aufgetragen. Ein Kontextübergang zwischen den beiden dargestellten Kontexten entspricht dabei einem Pfad auf der dargestellten Fläche, der die beiden Spitzen der Berge verbindet. Dies führt in der zeitlichen Betrachtung des Übergangs wiederum zu einer Verlaufskurve der Verlässlichkeit der Kontexterkennung, wie sie in Abbildung 4.13 dargestellt ist. Ziel des FCR-Systems ist es nun, dieses nichtlineare Modell des Verlaufs der Kontextverlässlichkeit, zu approximieren.

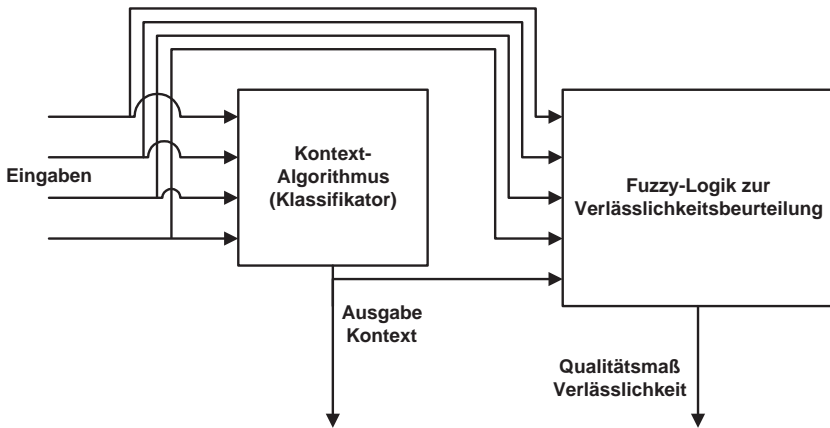


Abbildung 4.15. Schematische Architektur des FCR-Systems

Abbildung 4.15 zeigt eine schematische Darstellung der Architektur, des auf den einzelnen Artefakten einer ubiquitären Informationsumgebung lokalisierten FCR-Systems. Durch die Entkoppelung der Kontextklassifikation und der Bewertung der Qualität dieser Klassifikation, ist FCR unabhängig von der Art, der auf den Artefakten eingesetzten Kontextalgorithmen. Der Kontextalgorithmus wird als Blackbox betrachtet, die von außen durch das FIS beobachtet wird. Um eine korrekte Bewertung der Qualität der Ausgabe des Kontextalgorithmus zu gewährleisten, muss das FIS während der Entwicklung des Artefaktes an die konkreten Kontextalgorithmen angepasst werden, die in dem betreffenden Artefakt die Kontextererkennung implementieren. Hierzu kommen bekannte Techniken und Verfahren aus dem Bereich der *automatischen Identifikation von Fuzzy-Modellen* zum Einsatz.

Zu beachten ist hierbei, dass das eingesetzte Fuzzy-System nicht versucht einen unbekannten Klassifikator – den Kontextalgorithmus – nachzubilden, um das Qualitätsmaß zu bestimmen, sondern sowohl die Eingabe, als auch die Ausgabe des Kontextalgorithmus als Eingangsdaten erhält und diese durch die Zuordnung zu unscharfen Qualitätsklassen in Beziehung setzt.

Der Einsatz eines FIS zur Bewertung der Qualität der Kontexterkenennung eines Artefaktes hat zwei wichtige Vorteile gegenüber anderen Verfahren zur Beschreibung nichtlinearer Modelle: Erstens kann die Ausgabe eines FIS mit vergleichsweise geringen Ressourcenanforderungen berechnet werden [190] und zweitens kann die rechnerisch aufwändige Anpassung des FIS vollständig in die Entwicklungsphase eines Artefaktes verlegt werden [70]. Dabei wird das FIS für ein Artefakt wie in den folgenden Abschnitten gezeigt wird, automatisch auf einem leistungsstarken Computer erzeugt und optimiert.

4.7.2 Formalisierung: Generierung und Anpassung des Fuzzy-Systems während der Entwicklungsphase

Die korrekte Bewertung der Verlässlichkeit der Ausgabe eines Kontextalgorithmus erfordert die Erstellung (Identifikation oder Strukturidentifikation) und Anpassung (Konfiguration oder Parameteroptimierung) des zu verwendenden FIS. Die Identifikation des FIS ist Teil der Artefaktentwicklung; die Struktur des eingesetzten FIS ist abhängig von den, auf dem Artefakt implementierten Kontextalgorithmen und deren Klassifikationsleistungen. Da sich die Kontexterkenennungsalgorithmen eines Artefaktes zur Laufzeit nicht mehr ändern, sondern während seiner Entwicklung vollständig festgelegt werden, ist es sinnvoll, die Konfiguration des FIS ebenfalls während der Entwicklung des Artefaktes vorzunehmen.

Dieses Vorgehen hat den Vorteil, dass lediglich das endgültige FIS auf dem Artefakt implementiert werden muss. Dadurch ist FCR, obwohl es eine komplexe Funktionalität zur Verfügung stellt, vergleichsweise ressourcenschonend umsetzbar, wie in Kapitel 5 gezeigt wird.

Die Erstellung des FIS zur Qualitätsbeurteilung gliedert sich in zwei Phasen: in der ersten Phase wird ein initiales FIS identifiziert, welches dann in der zweiten Phase konfiguriert wird. Als Basis für diesen Erstellungsprozess dient die ANFIS-Architektur (Adaptive-Network-Based Fuzzy Inference System), die 1993 von Jang et al. vorgestellt wurde [87].

ANFIS-Architektur

Die ANFIS-Architektur interpretiert ein Takagi-Sugeno-Kang FIS [163, 159] (TSK-FIS), beziehungsweise dessen Parameter als neuronales Netzwerk (NN).

Auf diese Weise können verschiedene Lernmethoden, welche für die NN-Adaption entwickelt wurden, eingesetzt werden, um anhand von Trainings- und Prüfdaten, ein initiales FIS zu konfigurieren und damit seine Klassifikationsleistung zu optimieren.

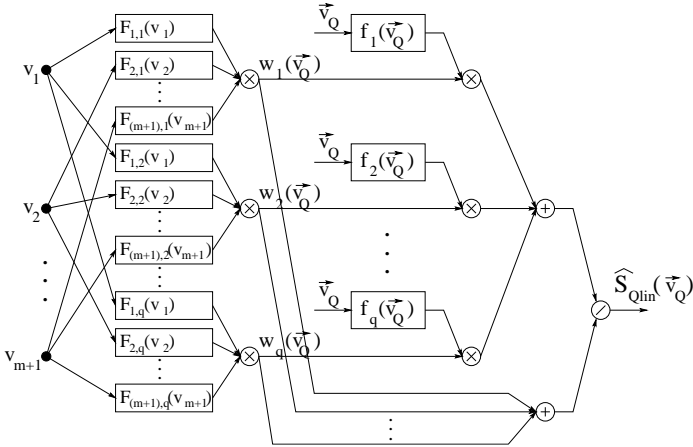


Abbildung 4.16. 4-Schicht-ANFIS-Architektur [20]

Konkret kommt in FCR ein ANFIS des 4-Schicht-Typs [88] zum Einsatz, wie es in Abbildung 4.16 zu sehen ist. Der Vektor $\vec{v_Q} = (v_1, v_2, \dots, v_{m+1})$ stellt die Eingaben des Qualitätssystems dar. Dies sind die m Eingaben des Qualitätsalgorithmus und seine Ausgabe (vgl. Abbildung 4.15). Die $F_{i,j}(v_i)$ bezeichnen die Zugehörigkeitsfunktionen des zu erstellenden FIS. Mit $w_j(\vec{v_Q})$ werden die Gewichte der Zugehörigkeitsfunktionen $F_{i,j}$ in Abhängigkeit von der Eingabe bezeichnet. $f_j(\vec{v_Q})$ bezeichnet die linearen Konsequenzen des TSK-FIS. Die Ausgabe des gesamten FIS wird mit $\hat{S}_{Qlin}(\vec{v_Q})$ bezeichnet.

Die Knoten der ersten Schicht dieses NN sind adaptive Knoten, welche die Zugehörigkeitsfunktionen $F_{i,j}$ enthalten und als Ausgabe, mit dem Betrag der Zugehörigkeit der Eingangsgröße feuern. Die zweite Schicht besteht aus festen Knoten \otimes , welche ihre Eingangssignale multiplizieren und deren Produkt ausgeben. Die dritte Schicht des NN besteht aus adaptiven Knoten, welche ihr Eingangssignal mit den Konsequenzparametern verbinden und diese zusammenfassen. Diese Schicht enthält einen weiteren Knoten \oplus , der seine Eingangssignale summiert.

In der letzten Schicht werden die Ausgaben der vorigen Schichten schließlich zusammengefasst und normalisiert.

Aufbauend auf dieser Architektur wird nun zunächst die Strukturidentifikation des FIS durchgeführt.

Strukturidentifikation eines TSK-FIS

Die Schwierigkeit bei der Bewertung der Verlässlichkeit der Kontexterkenennung mittels eines FIS besteht in der Erstellung der Zugehörigkeitsfunktionen, die dem Expertensystem zu Grunde liegen. Aufgrund der Komplexität einiger Kontexterkenner und ihrer zum Teil nicht trivial vorhersagbaren Ausgabe, zum Beispiel im Falle neuronaler Netze oder anderer so genannter *Blackbox-Verfahren*, können effiziente Zugehörigkeitsfunktionen vom Anwendungsentwickler kaum selbständig erstellt werden. Daher greift FCR auf eine automatische Identifikation wichtiger Struktureigenschaften des verwendeten FIS zurück. Andere strukturelle Parameter der eingesetzten FIS werden a priori festgelegt.

Zunächst wird nun auf die a priori festgelegten Strukturparameter der in FRC verwendeten FIS eingegangen, da diese die automatisch adaptierbaren Freiheitsgrade bedingen. Angesichts der Struktur des Modells des Verlaufes der Verlässlichkeit von Kontexten aus Abschnitt 3.8.4, werden als Basisfunktionen für die Erstellung der Zugehörigkeitsfunktionen F , Gaußfunktionen (4.3) gewählt.

$$F_{i,j}(v_i) := e^{-\frac{(v_i - \mu_{i,j})^2}{(2\sigma_{i,j}^2)}} \quad (4.3)$$

Gaußfunktionen werden hier als Basisfunktionen ausgewählt, da sie das Stone-Weierstrass-Theorem [175, 153] erfüllen [170, 171]. Dadurch ist es möglich mit der Hilfe von Gaußfunktionen, die durch das FIS darzustellende Zielfunktion beliebig gut zu approximieren. Außerdem lassen sich Gaußfunktionen besonders gut, mit der Hilfe automatischer Verfahren, durch die Anpassung ihrer Parameter σ und μ adaptieren.

Ein TSK-FIS besteht aus Regeln der Form:

$$\begin{aligned} \text{WENN } F_{1,j}(v_1) \text{ UND } F_{2,j}(v_2) \text{ UND } \dots \text{ UND } F_{m+1,j}(v_{m+1}) \text{ DANN } y = f_j(\vec{v_Q}) \\ \text{mit } i = 1, \dots, q \end{aligned} \quad (4.4)$$

In der unscharfen Logik werden die logischen Operatoren der scharfen Logik $\{\wedge, \vee, \neg\}$ durch die T-Norm T , die S-Norm S und die Negation N implementiert. Dies führt zum unscharfen Operatorsystem $\{T, S, N\}$.

Der nächste Schritt bei der Identifikation der Struktur des FIS besteht daher nun in der Festlegung konkreter Ausprägungen dieser unscharfen Operatoren. Hier wurde auf eine der in der Literatur gängigsten Implementierungen der unscharfen Operatoren zurückgegriffen [68]. Als T-Norm wird die Produktnorm $T_p(x, y) = xy$ gewählt, die zusammen mit der natürlichen Negation $N(x) = 1 - x$ die probabilistische Summe $S_p(x, y) = x + y - xy$ als S-Norm bedingt. Die unscharfe Implikation kann dann entsprechend als $I_p(x, y) = S_p(N(x), y) = 1 - x + xy$ dargestellt werden.

Nachdem die Zugehörigkeitsfunktionen und die unscharfen Operatoren definiert sind, wird als letzter Strukturparameter a priori die Norm festgelegt, welche bei der Berechnung der Ausgabe des FIS angewendet wird. Anders als bei Mamdani Fuzzy-Systemen [111], wird bei den hier eingesetzten TSK-FIS kein expliziter unscharfer Schluss gezogen und danach eine Defuzzifizierung durchgeführt, sondern direkt aus der Ausgabe der Zugehörigkeitsfunktionen $F_{i,j}$ und den linearen Konsequenzen f_j ein Ausgabewert berechnet. Dieser Berechnung wird im FCR-System die Summennorm L_1 zu Grunde gelegt. Die L_1 -Norm ist einfach zu berechnen und hat für die Qualität der Ausgabe des FIS keine signifikanten Nachteile gegenüber komplexeren Normen.

Im nächsten Schritt der Strukturidentifikation ist nun ein konkretes FIS zu erzeugen. Dazu muss die Anzahl der Zugehörigkeitsfunktionen $F_{i,j}$ festgelegt werden und es müssen initiale Werte ihrer Parameter σ und μ gefunden werden. Außerdem sind initiale Werte für die Gewichte $a_{i,j}$ der linearen Konsequenzen f_j (4.5) zu bestimmen.

$$f_j(\vec{v_Q}) := a_{1,j}v_1 + a_{2,j}v_2 + \dots + a_{m,j}v_m + a_{m+1,j}v_{m+1} + a_{m+2,j} \quad (4.5)$$

Für die Erzeugung des FIS wurde die Methode des *subtraktiven Clustering* gewählt, die von Stephen Chiu vorgeschlagen wurde [35, 36, 34, 37]. Der Vorteil dieses Verfahrens gegenüber anderen Ansätzen zur automatischen Erzeugung von FIS-Strukturen, liegt in der selbständigen Bestimmung der Anzahl der verwendeten Cluster und damit der benötigten Zugehörigkeitsfunktionen. Außerdem bestimmt subtraktives Clustering sehr gute Startwerte für die spätere Optimierung des FIS und kommt mit moderater Rechenzeit aus.

Hier wird nun ein kurzer Überblick über die Funktionsweise des subtraktiven Clustering gegeben. Eine detaillierte Beschreibung des Verfahrens kann [35] entnommen werden. Der Einsatz des Verfahrens zur automatischen Bestimmung von Zugehörigkeitsfunktionen für ANFIS-Systeme wird in [34] ausführlich diskutiert.

Subtraktives Clustering ist eine Abwandlung der *Mountain Methode* von Yager und Filev [178]. Beide Algorithmen sind in der Lage die Anzahl und Lage von Clustern automatisch zu bestimmen. Während beim Mountain Clustering die Eingabedaten mittels eines festen Gitters partitioniert werden, dessen Dimensionalität der Anzahl der Eingabevariablen entspricht, arbeitet subtraktives Clustering ohne eine a priori Partitionierung des Datenraumes. Mountain Clustering betrachtet alle Knoten des entstehenden Gitters als potentielle Clusterzentren und berechnet sukzessive für jeden dieser Knoten eine Potentialfunktion. Damit hängt der Rechenaufwand des Verfahrens von der Dimension des Eingaberaumes ab. Im Gegensatz dazu werden beim subtraktiven Clustering die Datenpunkte selbst als potentielle Clusterzentren betrachtet, wodurch der Aufwand mit der Größe des Trainingsdatensatzes skaliert. Dadurch ergibt sich in der praktischen Anwendung bereits bei einer Dimension > 3 des Eingaberaums, bei gleichem Rechenaufwand ein Auflösungsvorteil für das subtraktive Clustering.

Im ersten Schritt des subtraktiven Clusterings werden alle Eingabevektoren $\vec{v}_{Q,i}$ eines Trainingsdatensatzes $V_T = (\vec{v}_{Q,t}^T, \vec{v}_{Q,t+1}^T, \dots, \vec{v}_{Q,t+n}^T)$ normiert. Dazu wird hier die Maximum-Norm $\|\cdot\|_{\max}$ eingesetzt [20]:

$$\begin{aligned} \mathcal{V}_T &= \begin{pmatrix} \frac{1}{\|\vec{v}_{T,1}\|_{\max}} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\|\vec{v}_{T,m+1}\|_{\max}} \end{pmatrix} V_T \\ &= \begin{pmatrix} \frac{v_{1,t}}{\|\vec{v}_{T,1}\|_{\max}} & \cdots & \frac{v_{1,(t+n)}}{\|\vec{v}_{T,1}\|_{\max}} \\ \vdots & \ddots & \vdots \\ \frac{v_{(m+1),t}}{\|\vec{v}_{T,m+1}\|_{\max}} & \cdots & \frac{v_{(m+1),(t+n)}}{\|\vec{v}_{T,m+1}\|_{\max}} \end{pmatrix} \\ &= (v_{Q,t}^T, v_{Q,t+1}^T, \dots, v_{Q,t+n}^T) \end{aligned}$$

Dann werden die Potentiale der normierten Eingabevektoren nach Formel 4.6 berechnet. Dabei bezeichnet die Norm $\|\cdot\|_2$, die euklidische Distanz und r_a ist ein normalisierter Radius, welcher die Nachbarschaft eines Datenpunktes beschreibt.

Datenpunkte, deren Entfernung größer als r_a ist, beeinflussen das Potential nur sehr wenig.

$$P_i(\mathcal{V}_T) = \sum_{j=1, j \neq i}^{t+n} e^{-\alpha \|v_{Q,i} - v_{Q,j}\|_2^2} \quad \text{mit } \alpha = \frac{4}{r_a^2} \quad (4.6)$$

Sei $v_{Q,1}^*$ der Eingabevektor mit dem größten Anfangspotential P_1^* , dann wird dieser als erstes Clusterzentrum ausgewählt. Danach werden die Potentiale aller Datenpunkte nach Formel 4.7 angepasst.

$$P_i \leftarrow P_i - P_1^* e^{-\beta \|v_{Q,i} - v_{Q,1}^*\|_2^2} \quad \text{mit } \beta = \frac{4}{r_b^2} \quad (4.7)$$

Dabei ist r_b ein normalisierter Radius um $v_{Q,1}^*$, innerhalb dessen die Potentiale der dort befindlichen Datenpunkte eine signifikante Reduktion erfahren. Wie von Chiu in [34, 37] vorgeschlagen, wird in FRC ein Verhältnis von $\frac{r_a}{r_b} = 1,25$ und als Abbruchkriterium für die Berechnung neuer Cluster $\frac{P_k^*}{P_1^*} < 0,15$ verwendet.

Im nächsten Schritt werden nun die gefundenen Cluster direkt auf die zuvor definierten Zugehörigkeitsfunktionen $F_{i,j}(v_i)$ abgebildet. Dazu muss die Normierung der Eingabevektoren wieder aufgehoben werden, da das erstellte FIS später in Echtzeit auf den eingehenden Kontexten arbeiten soll und somit eine Normierung in der zeitlichen Dimension nicht möglich ist.

Die Mittelwerte $\mu_{i,j}$ der Gaußfunktionen werden von denen, durch das Clustering gefundenen Clusterzentren gebildet. Dabei gilt $\mu_{i,j} = \|\vec{v}_{T,i}\|_{\max} v_{i,j}^*$. Die Varianz der Gaußfunktionen $\sigma_{i,j}$, wird analog über den zuvor für das Clustering verwendeten Radius r_a definiert. Dabei sei $\sigma_{i,j}^{2*} = \frac{1}{2\alpha}$ [36, 34] und damit gilt dann für die Varianz der Zugehörigkeitsfunktion $\sigma_{i,j}^2 = \|\vec{v}_{T,i}\|_{\max} \sqrt{\sigma_{i,j}^{2*}}$. Damit sind die initialen Zugehörigkeitsfunktionen $F_{i,j}$ (4.3) definiert.

Nun werden die initialen Parameter der linearen Konsequenzen f_j (Formel 4.5) durch die Methode der kleinsten Quadrate bestimmt. Dazu werden die Parameter μ und σ der Zugehörigkeitsfunktionen als fest angenommen. Aus der Gleichung (4.4) und der Definition von T_p als Produktnorm folgt dann für die nichtlinearen Antezedenzen w_j :

$$w_j(\vec{v}_Q) := \prod_{i=1}^{m+1} F_{i,j}(v_i) \quad (4.8)$$

und für die normierten Antezedenzen \mathbf{w}_j :

$$\mathbf{w}_j(\vec{v}_Q) := \frac{w_j(\vec{v}_Q)}{\sum_{j=1}^q w_j(\vec{v}_Q)} \quad (4.9)$$

Die Ausgabe des FIS $\hat{S}_{Q_{lin}}(\vec{v}_Q)$ kann also geschrieben werden als:

$$\begin{aligned} \hat{S}_{Q_{lin}}(\vec{v}_Q) &= \sum_{j=1}^q \mathbf{w}_j(\vec{v}_Q) f_j(\vec{v}_Q) \\ &= \sum_{j=1}^q (\mathbf{w}_j(\vec{v}_Q) a_{1,j} v_1 + \dots + \mathbf{w}_j(\vec{v}_Q) a_{m+1,j} v_{m+1} + \mathbf{w}_j(\vec{v}_Q) a_{m+2,j}) \end{aligned} \quad (4.10)$$

Seien nun

$$\begin{aligned} \vec{a} &:= (a_{1,1}, \dots, a_{1,q}, a_{2,1}, \dots, a_{2,q}, \dots, a_{m+2,1}, \dots, a_{m+2,q}) \quad \text{und} \\ \vec{g} &:= (\mathbf{w}_1(\vec{v}_Q) v_1, \dots, \mathbf{w}_1(\vec{v}_Q) v_{m+1}, \mathbf{w}_1(\vec{v}_Q), \mathbf{w}_2(\vec{v}_Q) v_1, \dots, \mathbf{w}_2(\vec{v}_Q), \\ &\quad, \dots, \mathbf{w}_q(\vec{v}_Q) v_1, \dots, \mathbf{w}_q(\vec{v}_Q)) \end{aligned}$$

dann gilt:

$$\hat{S}_{Q_{lin}}(\vec{v}_Q) = \vec{g} \vec{a}^T \quad (4.11)$$

und für einen Trainingsdatensatz T :

$$\begin{pmatrix} \hat{S}_{Q_{lin}}(\vec{v}_{Q,t}) \\ \hat{S}_{Q_{lin}}(\vec{v}_{Q,t+1}) \\ \vdots \\ \hat{S}_{Q_{lin}}(\vec{v}_{Q,t+n}) \end{pmatrix} = \begin{pmatrix} \vec{g}_1 \\ \vec{g}_2 \\ \vdots \\ \vec{g}_{n+1} \end{pmatrix} \vec{a}^T = G_T \vec{a}^T \quad (4.12)$$

Sei der Fehler $\varepsilon(\vec{v}_{Q,k})$, definiert als Differenz der Ausgabe des FIS zu der gewünschten Ausgabe u_k :

$$\| \varepsilon(\vec{v}_{Q,k}) \|_2^2 := \| u_k - \hat{S}_{Q_{lin}}(\vec{v}_{Q,k}) \|_2^2 = \sum_{k=t}^{t+n} \left(u_k - \sum_{h=1}^{(m+2)q} g_{k,h} a_h \right)^2 \quad (4.13)$$

und der Gradient des Fehlers nach den Parametern der linearen Konsequenzen:

$$\frac{\partial \| \varepsilon(\vec{v}_{Q,k}) \|_2^2}{\partial a_p} = \sum_{k=t}^{t+n} \left(2 \left(u_k - \sum_{h=1}^{(m+2)q} g_{k,h} a_h \right) (-g_{k,p}) \right) = 0 \quad (4.14)$$

Transferiert man diesen Gradienten in Matrixschreibweise, kann er mittels einer Singulärwertzerlegung (Single Value Decomposition, SVD) [154] numerisch berechnet werden [20, 119]:

$$\begin{aligned}
 2(G_T \vec{a}^T - \vec{u})^T G_T &= 0 \\
 \vec{a}^T &= \left((G_T^T G_T)^{-1} G_T^T \right) \vec{u} \\
 \vec{a}^T &= (\mathbf{V} \Delta^T \mathbf{U}^T \mathbf{U} \Delta \mathbf{V}^T)^{-1} \mathbf{V} \Delta^T \mathbf{U}^T \vec{u} \quad \text{mit } G_T = \mathbf{U} \Delta \mathbf{V}^T \\
 \vec{a}^T &= \mathbf{V} \Delta^{-1} \mathbf{U}^T \vec{u}
 \end{aligned} \tag{4.15}$$

Parameteroptimierung eines TSK-FIS

Durch die Parameteridentifikation werden die, bei der Strukturidentifikation gefundenen, initialen nichtlinearen Parameter μ und σ und die linearen Parameter a optimiert. Für die Parameteridentifikation werden in der Literatur zwei grundsätzliche Herangehensweisen vorgeschlagen: zum einen, eine reine Optimierung aller Parameter durch Fehlerrückführung (engl. Backpropagation), zum anderen eine hybride Optimierung, bei der in jeder Lernepoche in einem „Rückwärtsschritt“, die nichtlinearen Parameter mittels Backpropagation optimiert werden und in einem „Vorwärtsschritt“ die Methode der kleinsten Fehlerquadrate eingesetzt wird, um die Parameter der linearen Konsequenzen zu optimieren.

Für die Anwendung in FCR wurde die hybride Parameteridentifikation gewählt, da bei dieser der Backpropagation-Algorithmus potentiell auf einem Suchraum kleinerer Dimension arbeitet und das Verfahren schneller konvergiert [88]. Das Backpropagation-Training des neuronalen Netzwerkes wurde der effiziente RPROP-Algorithmus [137] ausgewählt. Dieser hat den Vorteil, dass er leicht zu implementieren ist und seine Konvergenzgeschwindigkeit und Robustheit mit Gradientenverfahren zweiter Ordnung vergleichbar ist [120]. RPROP erfordert die Berechnung der ersten partiellen Gradienten der Fehlerfunktion nach den Parametern μ und σ . Sei:

$$\mathcal{E} = \|\varepsilon(\vec{v}_{Q,k})\|_2^2 \tag{4.16}$$

die Fehlerfunktion. Dann sind die gesuchten Gradienten:

$$\frac{\partial \mathcal{E}}{\partial \mu_{i,j}} \quad \text{und} \quad \frac{\partial \mathcal{E}}{\partial \sigma_{i,j}} \tag{4.17}$$

Für eine ausführliche Herleitung dieser Gradienten, sei an dieser Stelle zum Beispiel auf die Arbeiten von Martin Berchthold [20] und Manfred Männle [120] verwiesen.

RPROP optimiert die nichtlinearen Parameter ρ_j nach dem folgenden einfachen Schema:

$$\rho_j^\tau = \begin{cases} \rho_j^{\tau-1} - \lambda_j^\tau & \text{für } \frac{\partial \mathcal{E}^{\tau-1}}{\partial \rho_j} \cdot \frac{\partial \mathcal{E}^\tau}{\partial \rho_j} > 0 \text{ und } \frac{\partial \mathcal{E}^\tau}{\partial \rho_j} > 0 \\ \rho_j^{\tau-1} + \lambda_j^\tau & \text{für } \frac{\partial \mathcal{E}^{\tau-1}}{\partial \rho_j} \cdot \frac{\partial \mathcal{E}^\tau}{\partial \rho_j} > 0 \text{ und } \frac{\partial \mathcal{E}^\tau}{\partial \rho_j} < 0 \\ \rho_j^{\tau-2} & \text{für } \frac{\partial \mathcal{E}^{\tau-1}}{\partial \rho_j} \cdot \frac{\partial \mathcal{E}^\tau}{\partial \rho_j} < 0 \\ \rho_j^{\tau-1} & \text{sonst} \end{cases} \quad (4.18)$$

Die Schrittweite λ_j^τ , wird ebenfalls je nach Vorzeichen des Gradienten angepasst:

$$\lambda_j^\tau = \begin{cases} \eta^+ \cdot \lambda_j^{\tau-1} & \text{für } \frac{\partial \mathcal{E}^{\tau-1}}{\partial \rho_j} \cdot \frac{\partial \mathcal{E}^\tau}{\partial \rho_j} > 0 \\ \eta^- \cdot \lambda_j^{\tau-1} & \text{für } \frac{\partial \mathcal{E}^{\tau-1}}{\partial \rho_j} \cdot \frac{\partial \mathcal{E}^\tau}{\partial \rho_j} < 0 \\ \lambda_j^{\tau-1} & \text{sonst} \end{cases} \quad (4.19)$$

Untersuchungen ergaben gute Ergebnisse für die Wahl der Schrittweitenanpassungsfaktoren zu $\eta^+ = 1,2$ und $\eta^- = 0,5$ [183].

4.7.3 Berechnung und Setzen der Verlässlichkeit

Nach dem das FIS zur Qualitätsbestimmung, wie in Abschnitt 4.7.2 beschrieben erzeugt wurde, kann es auf das jeweilige Artefakt übertragen werden.

Die Ausgabe $\hat{S}_{Q_{lin}}$ des FIS stellt eine Klassifizierung der Eingabe, die ihrerseits aus der Ein- und Ausgabe des Kontextalgorithmus besteht, zu Qualitätsklassen dar. Wie zu Beginn des Abschnittes 4.7 erläutert wurde, approximiert $\hat{S}_{Q_{lin}}$ die nichtlineare Verlaufsfunktion der Kontextverlässlichkeit, wie sie in Abbildung 4.13 dargestellt wurde. Durch diese Approximation kann es vorkommen, dass $\hat{S}_{Q_{lin}}$ für bestimmte Eingabevektoren Rückgabewerte liefert, die außerhalb des Intervalls $[0, 1]$ liegen. Um aus dieser Zugehörigkeit zu Qualitätsklassen ein normiertes quantitatives Qualitätsmaß zu extrahieren, wird die Ausgabe des FIS $\hat{S}_{Q_{lin}}$ durch eine LR-Klassifikation auf das Intervall $[0, 1] \in \mathbb{R}$ projiziert. Dabei werden Werte, die außerhalb des Zielintervalls liegen, symmetrisch zu den Grenzen,

in dieses hinein projiziert. Das dabei verwendete spezielle Projektionsverfahren wurde von Martin Berchthold in [20] entwickelt.

Die LR-Klassifikation wendet konkateniert zwei Funktionen $\hat{L}_{\tilde{Q}Q_{in}}$ und $L_{Q\tilde{Q}}$ (Definition 4.20) auf das Ergebnis von $\hat{S}_{Q_{in}}$ an, die dieses zuerst auf eine Menge $\tilde{Q} = \{\tilde{0}, \tilde{1}\}$ von unscharfen Zahlen abbilden und dann die so berechnete Zugehörigkeit zu diesen unscharfen Zahlen wiederum auf eine scharfe Menge von Qualitätsklassen Q im Intervall $[0, 1] \in \mathbb{R}$ zurückführt. Das Ergebnis dieser Projektion ist das normierte Verlässlichkeitsmaß S_Q des FCR-Systems.

$$S_Q : \begin{cases} \mathbf{V}^{m+1} & \longrightarrow Q, \text{ mit } Q = \{[q_1], \dots, [q_P]\} \\ (v_1, v_2, \dots, v_{m+1}) & \longmapsto L_{Q\tilde{Q}} \circ \hat{L}_{\tilde{Q}Q_{in}} \circ \hat{S}_{Q_{in}}(v_1, v_2, \dots, v_{m+1}) \end{cases} \quad (4.20)$$

Die Berechnung von S_Q auf einem Artefakt ist also in zwei Schritte gegliedert. Bevor das FCR-System seine Berechnung beginnen kann muss der Kontextalgorithmus die Eingangsdaten – Sensorwerte oder Kontexte oder eine Kombination von beiden – klassifizieren und seinen Ausgabekontext erzeugen. Im ersten Schritt werden dann diese Eingangsdaten und der neu erzeugte Kontext vom Qualitätsaggregator an das FCR-Modul übergeben. Dort werden die Werte am Eingang des Verlässlichkeits-FIS bereitgestellt und daraus $\hat{S}_{Q_{in}}$ berechnet. Im zweiten Schritt wird dieses Ergebnis normiert und als quantitatives Verlässlichkeitsmaß S_Q in das Attributfeld FCR des neuen Kontextes geschrieben.

4.7.4 Auswertung der Verlässlichkeit

Die Auswertung des Attributes Verlässlichkeit von Kontexten wird vom Qualitätsfilter übernommen. Dazu extrahiert das Modul FCR den normierten Verlässlichkeitswert des Kontexten aus dem entsprechenden Attributfeld und übergibt ihn an den Filter.

Die Anwendung des Artefaktes kann den Qualitätsfilter mit einem Schwellwert für die akzeptable Qualität von Eingangskontexten konfigurieren und so die Einhaltung ihrer Qualitätsanforderungen durchsetzen. Der Vorteil dieses Verfahrens ist, dass der verarbeitenden Anwendung keine weiteren Informationen über die Entstehung ihrer Eingangskontexte bekannt sein müssen, um anhand des FCR-Verlässlichkeitsmaßes Kontexte zu filtern, da das Maß unabhängig von dem, zur Kontexterzeugung verwendeten Algorithmus ist.

4.8 Verwandtschaft von Kontexten (GRC)

Durch das Attribut Verwandtschaft von Kontexten (GRC) [188] wird das Qualitätsmanagementsystem in die Lage versetzt, die informationelle Abhängigkeit von Kontexten in einer ubiquitären Informationsumgebung dynamisch, während der Laufzeit der kontextsensitiven Anwendung, zu beurteilen.

CA	LOC	FCR	GRC
----	-----	-----	-----

Abbildung 4.17. Attribut Abhängigkeit von Kontexten (GRC)

Die hier vorgestellten Verfahren stellen einen völlig neuen Ansatz zur Beurteilung der Abhängigkeit von Kontexten dar. Die Umsetzung von GRC basiert auf Ideen, die der biologischen Genetik und dem Forschungsbereich der genetischen Algorithmen entlehnt wurden. Ziel dieses Kontextattributes ist es, ein Maß für die Verwandtschaft von Kontexten gemäß Definition 3.17 bereitzustellen, das auf effiziente Weise lokal durch die Artefakte einer ubiquitären Informationsumgebung ausgewertet werden kann und dessen Verwendung kein Wissen über die globale Verarbeitungstopologie der ubiquitären Informationsumgebung voraussetzt.

Kontextabhängigkeiten lassen sich, wie bereits gezeigt wurde, durch den Verarbeitungsgraph einer ubiquitären Informationsumgebung visualisieren. Der Verarbeitungsgraph selbst eignet sich allerdings nicht als Basis für die dynamische Analyse der Verwandtschaft von Kontexten durch die verarbeitenden Artefakte, da die Verarbeitungstopologie ubiquitärer Informationsumgebungen ständigen Änderungen unterworfen ist. Dadurch ist es nicht möglich, statische Verarbeitungsgraphen für eine Umgebung anzugeben und diese auf den Artefakten zu hinterlegen. Die dynamische Aktualisierung von Verarbeitungsgraphen auf den Artefakten ist aufgrund des hierdurch entstehenden Mehraufwandes ebenfalls nicht möglich.

Der Verarbeitungsgraph einer ubiquitären Informationsumgebung kann allerdings als Analysewerkzeug in kontrollierten Versuchsumgebungen eingesetzt werden. Hier gibt er wichtige Aufschlüsse über die Abhängigkeiten von kontextsensitiven Anwendungen und dient daher in dieser Arbeit als Grundlage für die theoretische Beurteilung der Leistungsfähigkeit der hier vorgestellten Verfahren.

Bei der Entwicklung von GRC auf die Nachbildung von biologischen Strukturen zurückzugreifen lag nahe, da mit dem biologischen Genom ein höchst effizientes Medium zur Handhabung von Verwandtschaftsinformationen existiert. In diesem Abschnitt werden daher auch die in der biologischen Genetik und für genetische Algorithmen gängigen Begriffe verwendet.

4.8.1 Genome von Kontexten

GRC etabliert ein Maß für die Verwandtschaft von Kontexten durch die Einführung eines Kontext-Genoms. Das Genom repräsentiert dabei die Identität jeder einzelnen Kontextinstanz und damit auch die Information, die durch diese verkörpert wird.

Immer, wenn ein neuer Kontext erster Ordnung produziert wird, wird auch ein neues Kontext-Genom generiert, welches mit diesem assoziiert wird. Werden Kontexte höherer Ordnung aus anderen Kontexten abgeleitet, werden die Genome der Quellkontexte rekombiniert und bilden so das Genom des neu abgeleiteten Kontextes. Auf diese Weise wird jedem Kontext in einer ubiquitären Informationsumgebung ein Genom zugeordnet. Eine Anwendung, die als Konsument Kontexte verarbeitet, kann nun die Genome der zu konsumierenden Kontexte vergleichen und so Rückschlüsse daraus ziehen, ob diese Kontexte in irgendeiner Weise verwandt oder unabhängig voneinander sind.

Sei C eine Kontextinstanz der Form $C = t, a, d$ und Γ_C das Genom dieser Instanz, dann bezeichnen die $\gamma_{C,i}$ einzelne Gene des Genoms von C .

$$\begin{aligned} C &:= \text{Kontextinstanz} \\ \Gamma_C &:= \text{Genom des Kontextes } C \\ \gamma_{C,i} &:= \text{Gen des Genoms } \Gamma_C \end{aligned}$$

Wird eine Kontextinstanz C erster Ordnung aus Sensordaten und Metainformationen abgeleitet, wird sie mit einem zufälligen Bit-Vektor der Länge l assoziiert. Dieser Bit-Vektor der Länge l repräsentiert das Genom Γ_C des Kontextes. Im nächsten Schritt wird das Genom Γ_C in n Gene γ_C sequenziert.

$$\Gamma_C := (\gamma_{C,1}, \gamma_{C,2}, \dots, \gamma_{C,n}) \quad (4.21)$$

Durch die Sequenzierung von Γ wird jedem Gen γ ein funktionaler *Genort*² zugeordnet, der durch den Index n adressiert wird. Anders als bei genetischen Algorithmen, kommt dem Genort eines Gens eine wichtige Bedeutung zu. Durch die Zuordnung eines bestimmten Gens zu einem Genort wird in GRC die Verwandtschaftsinformation gespeichert. Dieser Ansatz ist von dem biologischen Genom inspiriert. Auch hier kann anhand des Ortes und der Anordnung von Gensequenzen auf den Grad der Verwandtschaft von Genomen geschlossen werden [140]. Genetische Algorithmen verwenden diese Information dagegen nicht. Dort wird die Gesamtstruktur des Genoms mit einer Fitnessfunktion bewertet. Die Fitnessfunktion gibt dabei an, wie gut die durch das Genom repräsentierte Lösung eines Optimierungsproblems ist.

In GRC kommt dem Kontextgenom keine funktionale Aufgabe im Sinne einer optimierten Lösung für ein gegebenes Problem zu. Das Genom eines Kontextes repräsentiert nicht die Fitness oder die Qualität eines Kontextes, sondern identifiziert vielmehr die Information, die durch den Kontext dargestellt wird und in den Kontextdaten enthalten ist. Durch den Vergleich der genetischen Fingerabdrücke von mehreren Kontexten, kann GRC ein Maß für die Verwandtschaft der untersuchten Kontexte ausgeben, welches dann wiederum zur Beurteilung der Kontextqualität durch den Konsumenten herangezogen werden kann.

Die Leistungsfähigkeit von GRC wird dabei von verschiedenen Parametern beeinflusst. Dies ist zum einen die Anzahl n der Gene des Genoms und die Größe jedes einzelnen Gens. Die Größe der einzelnen Gene bestimmt die Anzahl verschiedener *Allele*³. Ein Allel entspricht dabei einem Wert, den das Gen annehmen kann. Legt man ein auf einem Bit-Vektor basierendes Genom zu Grunde, wird der Zusammenhang zwischen der Anzahl der Gene n , der Anzahl der Allele r und der Länge des Genoms l durch die folgende Formel beschrieben:

$$l = \frac{n \ln r}{\ln 2} \quad (4.22)$$

Durch die Erhöhung der Anzahl der Gene n kann die Auflösung des Verwandtschaftsgrades verfeinert werden. Erhöht man die Anzahl der Allele r jedes Gens,

²Der Genort oder *Locus* ist der Ort innerhalb eines Genoms, an dem ein bestimmtes Gen steht.

³Das Allel bezeichnet die Ausprägung, in der ein Gen an einem Genort vorliegt. Besteht das Kontextgen aus einem Bit-Vektor der Länge d , ergibt sich die Anzahl der möglichen Allele dieses Gens zu d^2 .

wird dadurch der systematische Fehler von GRC bei der Bestimmung des Verwandtschaftsgrades von Kontexten reduziert. Eine ausführliche Analyse der Einflüsse der Variation dieser Parameter von GRC findet sich in der Evaluierung des Systems in Kapitel 6.

Generell muss aber das Kontextgenom verglichen mit einem biologischen Genom extrem klein sein: Als Beispiel sei hier das einfache Darmbakterium *escherichia coli* angeführt. Dieses Bakterium verfügt über etwa 4.500 Gene, die in $4,6 \cdot 10^6$ Basenpaaren kodiert sind. Jedes dieser Gene trägt 4 Bit Information, was sich zu ca. 2,19 MB genetischer Information aufsummiert [2]. Dagegen kommen als Kontextgenome bei der Anwendung von GRC typischerweise Bit-Vektoren einer Größe von 50-200 Bytes zum Einsatz.

Die Reduktion der Größe der Kontextgenome ist möglich, da diese außer der Verwandtschaftsinformation keine weitere funktionale Information tragen. Im Gegensatz dazu speichert das biologische Genom eine Vielzahl weiterer Informationen, wie den „Bauplan“ und die Funktion des Organismus. Außerdem kann das Genom von Kontexten so ausgelegt werden, dass es den Anforderungen der ubiquitären Informationsumgebung in der es eingesetzt wird, gerecht wird. Dies betrifft vor allem die Anzahl an Generationen von Kontexten, die durch das Kontextgenom unterscheidbar sein sollen.

Werden Kontexte erster Ordnung produziert, so werden für dieses Kontexte neue Genome zufällig generiert. Werden dagegen Kontexte höherer Ordnung abgeleitet, so soll das Genom dieser Kontexte Rückschlüsse auf die informationelle Identität der Quellkontexte erlauben. Zu diesem Zweck wird das Genom von Kontexten höherer Ordnung durch Rekombination der Genome der Elternkontexte erzeugt. Der Prozess der Rekombination von Genomen wird bei genetischen Algorithmen *Crossover* genannt. Da bei GRC ein gängigen Crossover-Techniken ähnliches Verfahren zur Rekombination der Kontextgenome zum Einsatz kommt, wird auch hier der Begriff Crossover verwendet.

4.8.2 Probabilistic Multi-Site Crossover (PMSC)

Das Kontextgenom repräsentiert die Identität der Information, die ein Kontext transportiert. Um bei der Ableitung von Kontexten und der damit verbundenen Rekombination ihrer Kontextgenome, die für GRC notwendige Verwandtschaftsinformation zu generieren und zu erhalten, wurde im Rahmen dieser Arbeit ein

neues Crossover-Verfahren entwickelt, das speziell auf diese Anforderungen abgestimmt ist. Um die Unterschiede des hier vorgestellten Verfahrens zu gängigen Crossover-Verfahren, wie sie bei genetischen Algorithmen eingesetzt werden, zu illustrieren, ist es notwendig, kurz auf deren Funktion einzugehen.

Klassische Crossover-Verfahren

In genetischen Algorithmen kommt eine Vielzahl verschiedener Crossover-Verfahren zum Einsatz. Typischerweise werden die Genome hier allerdings nicht sequenziert, da die Größe der Gene und damit die Anzahl der möglichen Allele pro Gen für die Leistung genetischer Algorithmen keine Rolle spielt [62]. Typische Vertreter von gängigen Crossover-Verfahren sind „K-point Crossover“, „Cut and Slice“ sowie „Uniform Crossover (UX)“ und „Half Uniform Crossover (HUX)“. Eine genaue Beschreibung der Funktion dieser Verfahren und viele weitere Crossover-Verfahren für genetische Algorithmen finden sich in den Publikationen von J. H. Holland [80], K. A. De Jong [91] und W. Spears und V. Anand [151]. An dieser Stelle sollen nur typische Eigenschaften der Verfahren erwähnt werden, die sie für den Einsatz in GRC ungeeignet machen.

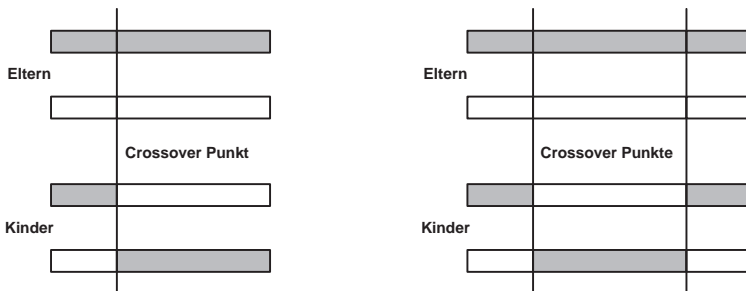


Abbildung 4.18. links: one point Crossover; rechts: two point Crossover

Generell sind Crossover-Verfahren für genetische Algorithmen darauf ausgelegt, eine $n:n$ -Verknüpfung von Genomen zu realisieren. Dabei werden n Elterngenome (oft $n = 2$) zu n Kindergenomen rekombiniert. Im Gegensatz dazu muss das Crossover-Verfahren von GRC eine $n:1$ -Verknüpfung von Kontextgenomen

realisieren, die der Verarbeitung von n Quellkontexten zu einem Zielkontext entspricht.

Der „K-point Crossover“ Operator zerschneidet das Elterngenom an k Stellen und tauscht die so entstandenen Teilgenome aus, um die Kindergenome zu erzeugen. Abbildung 4.18 zeigt zwei Varianten des K-point Crossover: den one point Crossover und den two point Crossover.

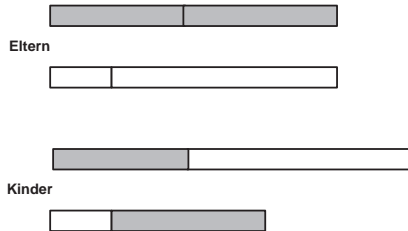


Abbildung 4.19. Cut and Slice Crossover

Der „Cut and Slice“ Operator wählt die Schnittpunkte in den Elterngenomen, wie Abbildung 4.19 zeigt, unabhängig voneinander. Dies führt zu einer variablen Länge der Genome.

UX und HUX sind probabilistische Crossover-Verfahren. Hier werden einzelne Bits zwischen den Elterngenomen ausgetauscht, um die Kindergenome zu erzeugen. Bei UX wird dabei eine feste Wahrscheinlichkeit zu Grunde gelegt, bei HUX werden genau 50% der nicht identischen Bits vertauscht.

Durch diese Operatoren wird das Elterngenom in einer Weise zerlegt, dass nicht garantiert werden kann, dass nach zwei Generationen noch genetisches Material der Großeltern vorhanden ist. Außerdem kann die Menge des genetischen Materials, das von den Eltern auf die Kinder übergeht, außer bei UX, nicht gesteuert werden. Bei „Cut and Slice“ und verwandten Verfahren führt zusätzlich die variable Länge der Genome zu Problemen.

Diese und andere bekannte Crossover-Verfahren für genetische Algorithmen sind darauf ausgelegt, sogenannte *Schemata* in den Genomen zu erhalten [62]. Schemata sind Muster in den Genomen, die Teile der Lösung eines Optimierungsproblems repräsentieren. Eine wichtige Eigenschaft, die Crossover-Verfahren, welche auf die Erhaltung von Schemata optimiert sind, für GRC ungeeignet

macht, ist, dass Schemata von dem Ort, an dem sie in einem Genom auftauchen, unabhängig sind. Dies führt dazu, dass die gängigen Crossover-Verfahren für genetische Algorithmen die Genorte einzelner Gene nicht erhalten. Dies ist aber notwendig, da GRC die Verwandtschaftsinformation über die Assoziation von Genen zu einem Genort kodiert.

Probabilistic Multi-Site Crossover (PMSC)

Probabilistic Multi-Site Crossover (PMSC) ist ein neuer Crossover-Operator, der speziell entwickelt wurde, um Verwandtschaftsinformationen in Kontextgenomen zu erhalten. PMSC ist ein probabilistisches Crossover-Verfahren, das auf den Genen des Kontextgenoms operiert. Es erhält bei seiner Anwendung den Genort vererbter Gene und erlaubt die Verknüpfung von n Elterngenomen. Außerdem erlaubt PMSC die Vererbungswahrscheinlichkeit so zu variieren, dass der Informationsanteil, der von Quellkontexten bei der Verarbeitung auf einen Zielkontext übergeht, nachgebildet werden kann. Dadurch kann das von GRC bereitgestellte Verwandtschaftsmaß direkt als Maß für die informationelle Abhängigkeit zwischen Kontexten interpretiert werden. Abbildung 4.20 zeigt die Grundfunktion des PMSC-Operators.

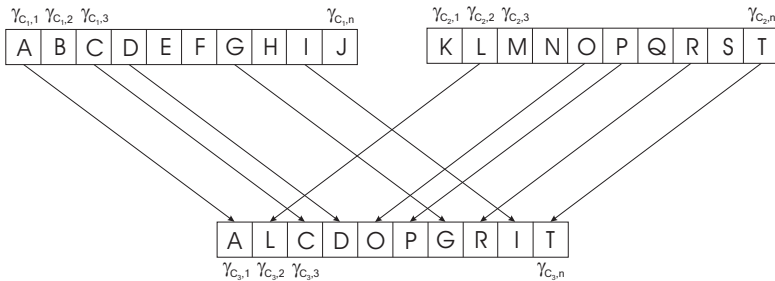


Abbildung 4.20. Vererbung von zwei Elterngenomenen

Bei der einfachsten Form der PMSC Anwendung werden zwei Genome Γ_{C_1} und Γ_{C_2} mit gleicher Vererbungswahrscheinlichkeit zu einem Genom Γ_{C_3} rekombiniert. In Abbildung 4.20 werden die verschiedenen Allele der Gene γ_C durch unterschiedliche Buchstaben dargestellt. PMSC geht bei seiner Anwendung schrittweise Genort für Genort durch die Elterngenome Γ_{C_1} und Γ_{C_2} und wählt zufällig

mit der vorgegebenen Vererbungswahrscheinlichkeit pro Genort eines der Eltern-gene aus und kopiert es an den entsprechenden Genort des Kindgenoms.

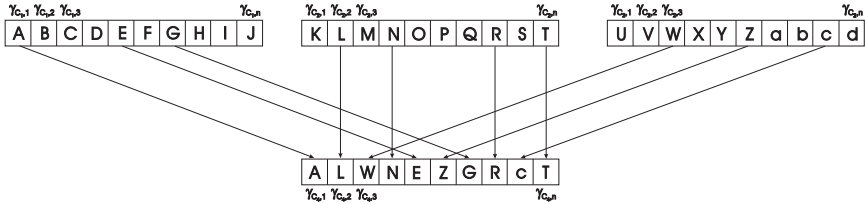


Abbildung 4.21. Vererbung von drei Elterngenomen

Abbildung 4.21 zeigt schematisch die Vererbung von drei Elterngenomen. Auch hier werden die Gene aller Eltern mit der gleichen Wahrscheinlichkeit an das Kindgenom weitergegeben. Im realen Einsatz muss die Wahl der Länge der Kontextgenome sicherstellen, dass von einer genügend großen Anzahl Eltern geerbt werden kann und trotzdem der Verwandtschaftsgrad über eine genügend Anzahl Generationen nachweisbar bleibt. Wie konkrete Werte für die Parameter von GRC festgelegt werden können, wird im Kapitel 5 Implementierung ausführlich diskutiert.

PMSC ist ein hoch effizientes Verfahren für die Implementierung auf Kleinstcomputern, wie sie heute die Basis für Artefakte in ubiquitären Informationsumgebungen bilden, da keine ressourcenintensiven Rechenoperationen nötig sind, um die Kontextgenome zu rekombinieren.

4.8.3 Berechnung des Verwandtschaftsgrades von Kontexten

Auf der Basis der so erzeugten Kontextgenome kann auf effiziente Weise der Verwandtschaftsgrad von Kontexten berechnet werden. Auch bei der Berechnung des Verwandtschaftsgrades von Kontexten kommen Algorithmen zum Einsatz, die möglichst wenig Ressourcen erfordern und eine geringe Laufzeit haben.

Der Verwandtschaftsgrad von Kontexten wird paarweise berechnet. Müssen die Abhängigkeiten von mehr als zwei Kontexten analysiert werden, resultiert dies in einer Verwandtschaftsmatrix. Bei der Berechnung des Verwandtschaftsgrades

werden die Genome der beiden zu vergleichenden Kontexte genweise analysiert. Dazu wird zunächst die Indikatorfunktion $f_i(\gamma_{C_1,i}, \gamma_{C_2,i})$ definiert:

$$f_i(\gamma_{C_1,i}, \gamma_{C_2,i}) := \begin{cases} 0 & : \gamma_{C_1,i} \neq \gamma_{C_2,i} \\ 1 & : \gamma_{C_1,i} = \gamma_{C_2,i} \end{cases} \quad (4.23)$$

Diese Funktion gibt eine „1“ zurück, wenn die Allele eines Genorts beider Kontexte C_1 und C_2 übereinstimmen. Stehen an einem Genort in beiden Kontexten verschiedene Allele eines Gens, liefert $f_i(\gamma_{C_1,i}, \gamma_{C_2,i})$ eine „0“ zurück.

Der Verwandtschaftsgrad $rel(C_1, C_2)$ wird dann berechnet, in dem die Ergebnisse der $f_i()$ über alle Genorte i aufsummiert werden und das Ergebnis gegen die Anzahl n der Gene des Genoms normalisiert wird:

$$rel(C_1, C_2) = \frac{1}{N} \sum_{n=1}^N f_n(\gamma_{C_1,n}, \gamma_{C_2,n}) \quad (4.24)$$

Diese Methode erlaubt eine schnelle und vor allem Ressourcen schonende Berechnung des Verwandtschaftsgrades von Kontexten. Das Ergebnis dieser Berechnung kann nun direkt als Anteil der Information interpretiert werden, den die Kontexte C_1 und C_2 gemeinsam haben. Der Verwandtschaftsgrad von Kontexten ist dabei, wie man leicht sieht, symmetrisch. Das heißt, es gilt: $rel(C_1, C_2) = rel(C_2, C_1)$.

Grundmuster der Vererbung

Die Funktion des Verwandtschaftsmaßes wird nun anhand von zwei Grundmustern der Vererbung illustriert. Das erste dieser Grundmuster, die stufenweise Vererbung, ist in Abbildung 4.22 schematisch dargestellt. Dieses Vererbungsmuster basiert auf einer Anzahl von Kontexten erster Ordnung. Aus zwei dieser Kontexte wird ein Kontext höherer Ordnung C_2 abgeleitet. In jedem weiteren Vererbungsschritt wird dann der resultierende Kontext und ein neuer Kontext erster Ordnung verknüpft. Auf diese Weise wird eine ununterbrochene Linie direkter Kontextableitungen erzeugt. Dabei kann beobachtet werden, wie in jedem weiteren Schritt der Verwandtschaftsgrad des erzeugten Kontextes im Vergleich zum Stammelternkontext C_1 abnimmt.

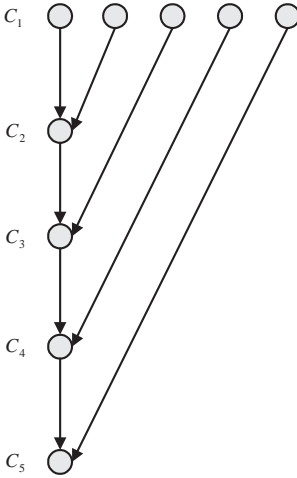


Abbildung 4.22. Stufenweise Vererbung

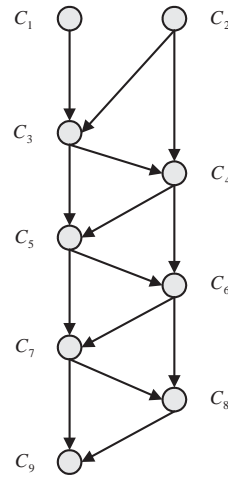


Abbildung 4.23. Mehrfachvererbung

Abbildung 4.24 zeigt das gemittelte Verwandtschaftsmaß einer stufenweisen Vererbung über 10 Generationen von Kontexten. Dem hier gezeigten Versuch lag ein Genom von 100 Genen mit 256 Allelen zu Grunde. Die Grafik zeigt eine Mittelung der Ergebnisse über 10.000 Versuchsdurchläufe. Auf der y-Achse ist der Grad der Verwandtschaft aufgetragen, auf der x-Achse die Generationen. Die Kurve repräsentiert die $rel(C_1, C_i)$.

Wie zu erwarten war, fällt der Verwandtschaftsgrad von einem Startwert von $rel(C_1, C_2) \approx 0.5$ mit jeder weiteren Generation. Dabei wird der Anteil des genetischen Materials von C_1 in jedem Schritt etwa halbiert. Dies hat zur Folge, dass das Verwandtschaftsmaß, welches durch GRC bereitgestellt wird, immer einen festen Wert für einen bestimmten Verwandtschaftsgrad liefert, der nicht von der Parameterkonfiguration von GRC abhängt. Lediglich die Auflösung und damit die Genauigkeit, mit der der Verwandtschaftsgrad in höheren Generationen bestimmt werden kann, hängt von den Parametern ab.

Das zweite Grundmuster, die Mehrfachvererbung, ist in Abbildung 4.23 dargestellt. Ausgehend von zwei unabhängigen Elternkontexten C_1 und C_2 , werden neue Kontexte abgeleitet, indem ein Kontext mit einem seiner direkten Vorfahren

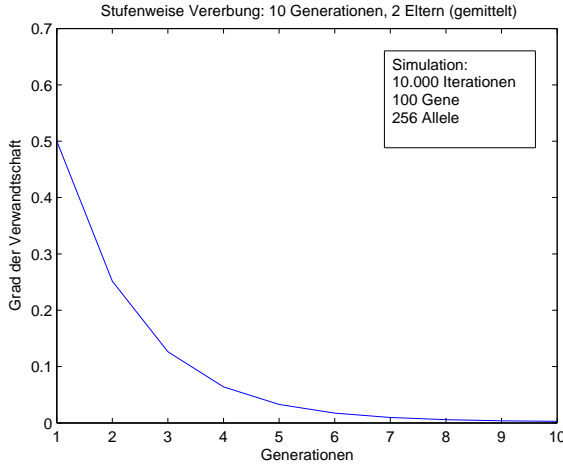


Abbildung 4.24. Stufenweise Vererbung über 10 Generationen

verknüpft wird. Im Gegensatz zur stufenweisen Vererbung werden hier also keine neuen Informationen der Vererbungskette hinzugefügt.

In Abbildung 4.25 ist das gemittelte Verwandtschaftsmaß einer Mehrfachvererbung über 10 Kontextgenerationen zu sehen. Für diesen Versuch wurde die gleiche Konfiguration, wie für das Beispiel der stufenweisen Vererbung, gewählt. Der gezeigte Graph repräsentiert wieder die $rel(C_{i-1}, C_i)$, beginnend mit $i = 3$.

Man sieht, dass nun beginnend mit dem Ausgangswert $rel(C_2, C_3) \approx 0.5$ der Verwandtschaftsgrad der untersuchten Kontexte mit jeder weiteren Generation ansteigt. Dies ist darauf zurückzuführen, dass keine neuen Informationen in den Vererbungsprozess einfließen. Dadurch werden sich bei diesem Vererbungsmuster die Genome aufeinanderfolgender Generationen immer ähnlicher: das Genom konvergiert.

4.8.4 Systematische Überschätzung des Verwandtschaftsgrades

GRC ist ein probabilistisches Verfahren, um den Grad der Verwandtschaft von Kontexten in einer ubiquitären Informationsumgebung zu bestimmen. Das bedeu-

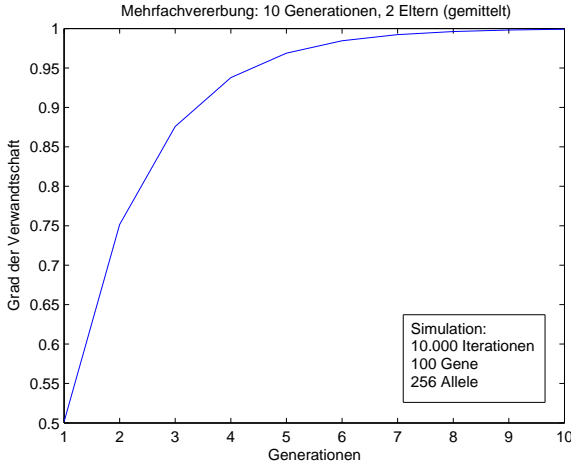


Abbildung 4.25. Mehrfachvererbung über 10 Generationen

tet, GRC liefert nicht den realen Verwandtschaftsgrad zweier Kontexte, sondern ist ein effizienter Schätzer für diesen Wert. Die Qualität dieses Schätzers hängt, wie bereits erwähnt, von der Wahl der GRC-Parameter ab. Im Folgenden wird eine grundlegende Eigenschaft von GRC untersucht: die systematische Überschätzung des Verwandtschaftsgrades. Diese führt zu einem systematischen Fehler, der allerdings durch GRC bei der Filterung von Kontexten, bewusst ausgenutzt wird.

Die endliche Größe eines Gens γ und die dadurch beschränkte Anzahl Allele r , führt dazu, dass bei der zufälligen Generierung von Kontextgenomen zufällig gleiche Allele an einem Genort entstehen können, sogenannte Dubletten. Werden die Bit-Vektoren der Genome basierend auf einer Gleichverteilung erzeugt, gilt für die Wahrscheinlichkeit $P(i)$, dass i Dubletten in zwei zufällig erzeugten Genomen, mit n Genen und r Allelen, zu finden sind:

$$P(i) = \left(\frac{1}{r}\right)^i \left(1 - \frac{1}{r}\right)^{n-i} \binom{n}{i} \quad (4.25)$$

Diese Dubletten können zu einem Fehler bei der Bestimmung des Verwandtschaftsgrades von Genomen führen. Sind bei der Auswertung des Verwandt-

schaftsgrades zweier Genome Allele eines Genorts gleich, kann nicht entschieden werden, ob diese zufällig aufgrund des Generierungsprozesses der Genome entstanden sind, oder bei der Vererbung durch die Anwendung des PMSC-Operators.

Die Wahrscheinlichkeit, bei der Bestimmung des Verwandtschaftsgrades einen Fehler zu machen, ist nach oben beschränkt. Diese obere Schranke des Fehlers beschreibt den Fall, dass alle Dubletten, die in den Elterngenomen eines Kontextes vorkommen, auch tatsächlich zu einem Fehler führen. Der Dubletten-Fehler an sich ist dadurch definiert, dass die Summe der Verwandtschaftsgrade eines Kontextes zu einer vollständigen Elterngeneration größer als 1 ist.

In der Realität kann es vorkommen, dass ein Dubletten-Fehler während des Vererbungsprozesses verschwindet. Es ist sogar so, dass die Wahrscheinlichkeit, dass ein Fehler Auswirkungen auf das Ergebnis der Bestimmung des Verwandtschaftsgrades von zwei Genomen hat, sinkt, je komplexer der Prozess wird. Die Fehlerwahrscheinlichkeit selbst ist aber letztendlich vom konkreten Vererbungsprozess und damit vom Verarbeitungsprozess abhängig. Sie kann in allgemeiner Form ohne Wissen über den erzeugenden Prozess nicht angegeben werden. Daher muss sich bei der Anwendung von GRC auf die Betrachtung der oberen Schranke dieses Fehlers beschränkt werden.

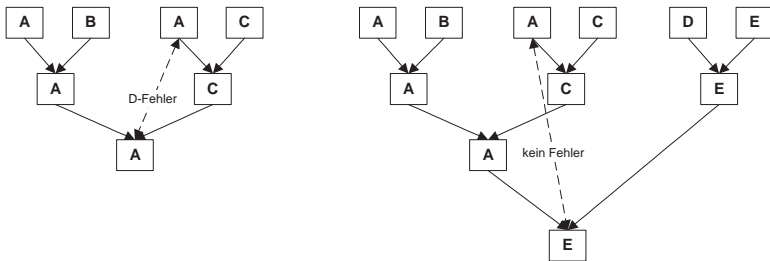


Abbildung 4.26. Dubletten-Fehler können durch Vererbung verschwinden

Wie Dubletten-Fehler innerhalb des Vererbungsprozesses verschwinden können, soll an einem einfachen Beispiel illustriert werden. Dabei wird nur ein einziger Genort innerhalb der beteiligten Kontextgenome betrachtet. Abbildung 4.26 zeigt einen Vererbungsprozess der Gene an diesem Genort. In der linken Bildhälfte tritt ein Dubletten-Fehler in der letzten Generation auf, da hier das vererbte Allel zu einem Fehler bei der Berechnung der Verwandtschaftsgrade der Aus-

gangskontexte und des Kontextes der letzten Generation führen kann. Wird der Vererbungsprozess, wie in der rechten Bildhälfte fortgesetzt, kann in der nächsten Generation das fehlerauslösende Dubletten-Allel verschwinden, weil es nicht weitervererbt wird. Der maximale Fehler kann nur auftreten, wenn in allen Fällen ein Allel, das potentiell zu einem Fehler führt, auch weitervererbt werden würde. Zu beachten ist, dass dieses Beispiel nur einen einzigen Genort zeigt. Daher kann aus Abbildung 4.26 alleine nicht abgelesen werden, ob der gezeigte Sachverhalt wirklich zu einem Fehler führt.

Die obere Schranke der Fehlerwahrscheinlichkeit für Dubletten-Fehler berechnet sich wie folgt: ein Dubletten-Fehler kann auftreten, wenn bei der zufälligen Generierung von Genomen Dubletten entstehen. Die obere Schranke dieses Fehlers wird erreicht, wenn alle Dubletten in allen beteiligten Genomen zu einem Fehler führen; also genau dann, wenn keine Dubletten während des Vererbungsprozesses verlorengehen. Dies entspricht dann der Wahrscheinlichkeit $P(err)$, dass an einem Genort, der Genome aller beteiligten Eltern m , mindestens eine Dublette zu finden ist.

Die Berechnung erfolgt nun über das Komplement $P^c(err)$ obiger Aussage. $P^c(err)$ beschreibt die Wahrscheinlichkeit, dass keines der Elterngenome eine Dublette am betrachteten Genort hat. Das heißt, alle Elterngenome besitzen am untersuchten Genort unterschiedliche Allele eines Gens:

$$P^c(err) = \frac{r}{r} \cdot \frac{r-1}{r} \cdot \frac{r-2}{r} \cdots \frac{r-m+1}{r} = \prod_{j=0}^{m-1} \frac{r-j}{r} \quad (4.26)$$

Dies führt zur Formel für die obere Schranke der Fehlerwahrscheinlichkeit $P(err)$:

$$P(err) = 1 - P^c(err) = 1 - \prod_{j=0}^{m-1} \frac{r-j}{r} = 1 - \frac{r!}{r^m(r-m)!} \quad (4.27)$$

$P(err)$ beschreibt die Wahrscheinlichkeit, dass an einem Genort eines Genoms eine zufällige Dublette auftritt. Unter der Annahme des schlimmsten Falles, führt jede dieser Dubletten auch zu einem Fehler. Damit ist $P(err)$ auch eine obere Schranke der Fehlerwahrscheinlichkeit. Betrachtet man ein Genom bestehend aus

n Genen, muss man in diesem Genom folglich $n \cdot P(err)$ Dubletten-Fehler erwarten. Tritt ein Dubletten-Fehler auf, beeinflusst dieser das Ergebnis der Berechnung des Verwandtschaftsgrades, indem sich für jede auftretende Dublette, die zum Fehler führt, der Verwandtschaftsgrad um $\frac{1}{n}$ erhöht (siehe Berechnung von $rel(C_1, C_2)$ in Formel 4.23 und 4.24). Für den Erwartungswert des Fehlers $E(err)$, bei Eintreten des schlimmsten Falles bedeutet dies:

$$E(err) = n \cdot P(err) \cdot \frac{1}{n} = P(err) \quad (4.28)$$

Damit bildet $P(err) = E(err)$ eine obere Schranke des Fehlers bei beliebigen Vererbungsprozessen, an denen m unabhängige Elterngenome beteiligt sind. Abbildung 4.27 zeigt den Verlauf von $P(err)$ für eine variable Anzahl von an einem Vererbungsprozess beteiligten Elterngenomen im Überblick.

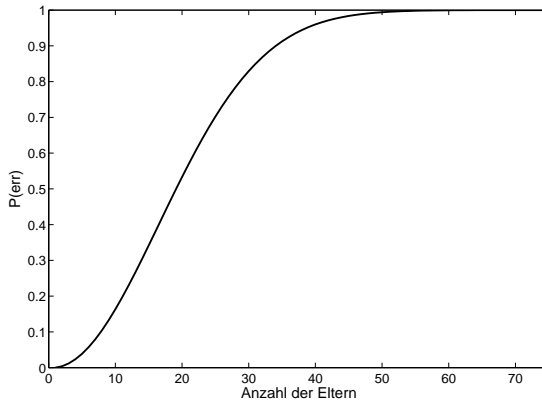


Abbildung 4.27. $P(err)$ mit 256 verschiedenen Allelen (75 Eltern)

Die Grafik beruht auf einer Konfiguration mit 256 verschiedenen Allelen. Abbildung 4.28 zeigt einen Ausschnitt obiger Abbildung 4.27. Hier ist der, für reale Anwendungen relevante Verlauf von $P(err)$ für bis zu 10 Elterngenome dargestellt.

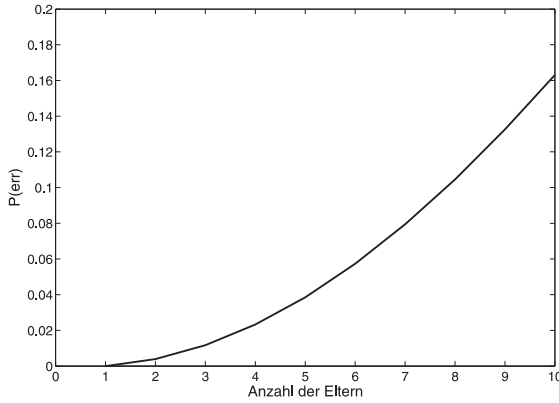


Abbildung 4.28. $P(err)$ mit 256 verschiedenen Allelen (10 Eltern)

Da über die Verteilung bestimmter Vererbungsprozesse (oder Prozessklassen), die zu einem bestimmten Fehler führen, keine weiteren statistischen Annahmen getroffen werden können, muss hier das Prinzip der maximalen Entropie [89] zu Grunde gelegt werden. Für bestimmte Prozesse⁴ kann allerdings ein Schätzer angegeben werden, der aus einem von GRC berechneten Verwandtschaftsgrad $rel(C_1, C_2) = \beta$ auf den tatsächlichen Verwandtschaftsgrad der Kontexte α schließen lässt.

Als Beispiel für einen solchen Schätzer soll der folgende elementare Vererbungsprozess betrachtet werden: Aus zwei Elterngenomen wird ein Kindgenom abgeleitet. Dieser einfachste Vererbungsprozess zeichnet sich dadurch aus, dass eine Dublette in den Elterngenomen zwangsläufig zu einem Fehler bei der Berechnung der Verwandtschaftsgrade des Kindes zu einem der Elternteile führt. Bei komplexeren Vererbungsprozessen ist dies dagegen nicht garantiert.

Grundlage für den Schätzer ist eine Analyse des Vererbungsprozesses mittels eines Entscheidungsbaumes. Dazu wird zunächst ein einzelner Genort der Genome betrachtet. Die Wahrscheinlichkeit, dass an einem Genort bei zwei zufällig

⁴Bestimmte Prozesse meint hier Prozesse, die bekannt sind und sich daher so formulieren lassen, dass die Fehlerwahrscheinlichkeiten in jedem Schritt modelliert werden können.

erzeugten Genomen eine Dublette auftritt ist $\frac{1}{r}$, wobei r die Anzahl der möglichen Allele eines Gens bezeichnet. Die Wahrscheinlichkeit, dass das Allel eines Gens weiter vererbt wird sei α . Dann berechnet sich die Wahrscheinlichkeit, dass ein Genort des Kindes und eines Elternteiles das gleiche Allel eines Gens hält, zu:

$$1 - \left(\left(1 - \frac{1}{r} \right) (1 - \alpha) \right) = \alpha + \frac{1 - \alpha}{r}$$

Daraus lässt sich nun der Schätzer $P(\alpha|\beta)$ ableiten, der die Wahrscheinlichkeit für einen realen Verwandtschaftsgrad von α unter der Bedingung, dass GRC einen Verwandtschaftsgrad von β zurückgeliefert hat, berechnet:

$$\alpha_\sigma := \alpha + \frac{1 - \alpha}{r}$$

$$P(\alpha|\beta) = \alpha_\sigma^{N\beta} (1 - \alpha_\sigma)^{N(1-\beta)} \binom{N}{\beta N} \quad (4.29)$$

Abschließend wird nun aus den obigen Ergebnissen, der zu erwartende Fehler in der Berechnung des Verwandtschaftsgrades zu einem beliebigen Elterngenom in einem Vererbungsprozess mit m beteiligten Elterngenomen $E(rel_{err})$ hergeleitet. Zuerst wird dazu die zu erwartende Anzahl von Dubletten in m zufällig erzeugten Elterngenomen mit n Genen und je r Allelen betrachtet. An jedem Genort eines Elterngenoms wird das dortige Gen mit einem zufälligen Allel belegt. Für die Anzahl der zu erwartenden Dubletten an diesem Genort gilt dann:

$$0 + \frac{1}{r} + \frac{2}{r} + \dots + \frac{m-1}{r} = \sum_{i=1}^m \frac{i-1}{r}$$

Dieser Vorgang wird für alle n Gene der Elterngenome wiederholt. Dabei kann davon ausgegangen werden, dass die entstehenden Dubletten, auf die m Elterngenome gleich verteilt sind:

$$\frac{n \sum_{i=1}^m \frac{i-1}{r}}{m}$$

Obige Formel gibt die zu erwartende Anzahl Dubletten zu einem beliebigen Elterngenom an. Tritt ein Dubletten-Fehler auf, beeinflusst jeder dieser Fehler

das Ergebnis der Berechnung des Verwandtschaftsgrades durch eine Erhöhung des Ergebnisses um $\frac{1}{n}$. Nimmt man nun wieder an, dass ohne Rücksicht auf den tatsächlichen Vererbungsprozess, jede Dublette zu einem Fehler führen würde, ergibt sich der zu erwartende Fehler $E(rel_{err})$ in der Berechnung des Verwandtschaftsgrades zu einem beliebigen Elterngenom, zu:

$$E(rel_{err}) = \frac{1}{m} \sum_{i=1}^m \frac{i-1}{r} \quad (4.30)$$

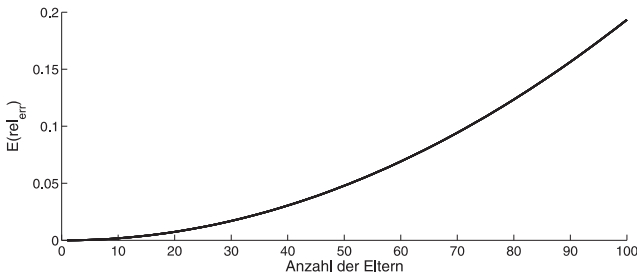


Abbildung 4.29. $E(rel_{err})$ mit 256 verschiedenen Allelen, bis 100 Eltern

Abbildung 4.29 zeigt, wie sich dieser Fehler bei einer steigenden Anzahl von Elterngenomen auf den Verwandtschaftsgrad auswirkt.

4.8.5 Setzen des Kontextgenoms

Beim Setzen des Verwandtschaftsgrades durch das Modul GRC müssen die beiden grundlegenden Fälle von Kontexten erster Ordnung und Kontexten höherer Ordnung unterschieden werden. Bei Kontexten erster Ordnung wird das Kontextgenom zufällig für jede Kontextinstanz erzeugt. Dagegen wird das Kontextgenom von Kontexten höherer Ordnung aus den Genomen der Elternkontexte generiert, aus denen diese abgeleitet werden.

Werden Kontexte erster Ordnung produziert, generiert das Modul GRC ein zufälliges Genom mit den vorgegebenen Parametern für die Genomlänge und Anzahl der Allele. Dieses Genom wird in das Attributfeld GRC des neuen Kontextes kopiert.

Werden Kontexte höherer Ordnung produziert, erhält das Modul GRC die Genome der Elternkontexte aus dem Qualitätsaggregator. Zusätzlich benötigt es die Information, in welchem Verhältnis die Genome der Elternkontexte auf das Genom des neuen Kontextes übergehen sollen. Dies wird von der Anwendung festgelegt und ebenfalls über den Qualitätsaggregator bereitgestellt. Das Modul GRC wählt dann entsprechend dieses Verhältnisses zufällig Gene der Eltern aus, deren Allele auf das Kindgenom übergehen.

Wird ein Kontext höherer Ordnung aus nur einem Elternkontext erzeugt, wird das Kindgenom aus dem Genom dieses Elternkontextes und einem zufällig erzeugten neuen Genom gebildet, welches die durch die Verarbeitung hinzukommende Sensor- oder Metainformation repräsentiert. Dabei wird wieder von der Anwendung über den Qualitätsaggregator vorgegeben, in welchem Verhältnis die beiden Genome auf das Kind übergehen.

Ist so das neue Kontextgenom zusammengestellt, wird es in das entsprechende Attributfeld des Kontextes kopiert. Gleichzeitig wird das neue Genom in den so genannten Historienspeicher des Artefaktes kopiert. Dieser dient dem Artefakt dazu, die Genome eingehender Kontexte mit den Genomen von Kontexten, welche das Artefakt selbst produziert hat, vergleichen zu können.

4.8.6 Auswertung des Kontextgenoms

Bei der Auswertung des Kontextgenoms muss unterschieden werden, ob das konsumierende Artefakt Kontexte fusionieren will oder Eingangskontexte zur Ableitung neuer Kontexte verwendet. In beiden Fällen werden die zu konsumierenden Kontexte zunächst im Verarbeitungspuffer der Kontextmanagementschicht gesammelt. Wenn dann genügend Kontexte vorhanden sind und dadurch die Ableitung oder Fusion möglich wird, werden die gepufferten Kontexte weiterverarbeitet. Die Genome der Kontexte werden paarweise auf ihren Verwandtschaftsgrad überprüft. Die so entstehende symmetrische Verwandtschaftsmatrix (siehe 4.31) wird zur Kontextauswahl an den Qualitätsfilter weitergegeben.

$$\begin{pmatrix} 1 & rel(C_1, C_2) & \cdots & & rel(C_1, C_n) \\ rel(C_2, C_1) & 1 & & & \\ \vdots & & \ddots & & \vdots \\ & & & 1 & rel(C_{n-1}, C_n) \\ rel(C_n, C_1) & \cdots & rel(C_n, C_{n-1}) & & 1 \end{pmatrix} \quad (4.31)$$

Zusätzlich zu den Genomen der Eingangskontexte wird eine artefaktinterne Liste der letzten, selbst produzierten Kontextgenome in die Analyse mit einbezogen, der Historienspeicher. Diese Liste enthält die Genome einer konfigurierbaren Anzahl von Kontexten, welche das Artefakt zuletzt produziert hat. Sie wird benutzt, um die Bildung von Kreisen in der Kontextverarbeitung effizient handhaben zu können. Dies ist besonders notwendig im Falle rein fusionierender Artefakte, da diese Kontexte des gleichen Typs produzieren, den sie auch konsumieren.

Der Qualitätsfilter ist dann für die abschließende Auswahl der zu verarbeitenden Kontexte verantwortlich.

4.9 Der Qualitätsfilter

Der Qualitätsfilter verwaltet die Qualitätsanforderungen der Anwendung an Kontexte, die weiterverarbeitet werden sollen. Als Eingabe erhält der Qualitätsfilter Schwellwerte oder Wertebereiche innerhalb derer sich die Attributwerte eines Kontextes bewegen müssen. Liegen die Attributwerte außerhalb der vorgegebenen Bereiche, wird der betreffende Kontext verworfen. Auf diese Weise werden der semantischen Kontextverarbeitung nur Kontexte zugeführt, welche die von der Anwendung vorgegebenen Qualitätsanforderungen erfüllen.

Zur Erhöhung der Flexibilität und um Qualitätsanforderungen in einer feineren Granularität formulieren zu können, unterstützt der Qualitätsfilter die logische Verknüpfung von Anforderungen. Auf diese Weise können logische Regeln über den zur Verfügung stehenden Attributen formuliert werden, welche die Qualitätsanforderungen der Anwendung repräsentieren.

Die Aufgabe des Qualitätsfilters ist es, diese Regeln auszuwerten und sie auf die, von den zuvor besprochenen Modulen, vorverarbeiteten Kontextattribute anzuwenden.

4.9.1 Filterung: CA

Auf das Attribut Kontextalter wendet der Qualitätsfilter einfache Schwellwertfilter an. Der Anwendungsentwickler kann angeben, in welchen zeitlichen Bereichen relativ zum aktuellen Zeitpunkt, Kontextalter und Ableitungsdauer liegen dürfen, damit die Anwendung Kontexte akzeptieren kann.

Neben den direkten Auswirkungen auf die Kontextqualität, in Bezug auf das Kontextalter, kann durch die Filterung des Kontextalters und der Ableitungsdauer die Bildung von Kreisen in der Verarbeitung von Kontexten auf einer heuristischen Basis bis zu einem gewissen Grad vermieden werden. Das Kontextgenom kann aber durch diese eher einfache Methode nicht in seiner Funktionalität und Präzision ersetzt werden.

4.9.2 Filterung: LOG

Die Filterung auf das Attribut räumliche Herkunft erfolgt durch die Angabe von RAUM-konformen Interessensbereichen. Diese Interessensbereiche werden vom Anwendungsentwickler vorgegeben und können relativ zur aktuellen Lokation des Artefaktes formuliert werden.

Die eigentliche Filterung der Kontexte geschieht dann mittels der Auswertung der durch den Interessensbereich induzierten Inklusionsrelation: Liegt die räumliche Herkunft eines Eingangskontextes innerhalb eines der Interessensbereiche eines Artefaktes, wird der Kontext akzeptiert, ansonsten wird er verworfen.

4.9.3 Filterung: FCR

Zur Filterung auf das Kontextattribut Verlässlichkeit wird, ähnlich wie beim Kontextalter, ein einfacher Schwellwertfilter eingesetzt. Der Anwendungsentwickler kann eine Grenze angeben, ab der Kontexte mit niedrigerer Verlässlichkeit abgewiesen werden.

Das Attribut FRC kodiert die Verlässlichkeit eines Kontextes in einem einzigen Wert, der als Abstand der Kontextinstanz zum Mittelpunkt der Kontextklasse interpretiert werden kann. Daher muss auch die akzeptable Verlässlichkeitsschwelle individuell für jede Anwendung und jeden konsumierten Kontext bestimmt werden.

4.9.4 Filterung: GRC

Im Gegensatz zu den übrigen Attributen stellt die Verwandtschaft von Kontexten eine Beziehung zwischen den Kontexten her, die ein Artefakt konsumiert. Daher ist die Filterung von Kontexten auf der Basis der Verwandtschaftsinformation etwas anspruchsvoller in Bezug auf die Entscheidung, welche Kontexte abgelehnt und welche akzeptiert werden.

Im allgemeinen Fall wird dem Qualitätsfilter vom Modul GRC eine Matrix mit den paarweisen Verwandtschaftsgraden der Eingangskontexte zur Verfügung gestellt. Gegebenenfalls enthält diese Matrix auch die Verwandtschaftsgrade von Eingangskontexten zu Kontexten, welche von diesem Artefakt zuvor produziert wurden (siehe Abschnitt 4.8.6). Der Qualitätsfilter muss nun anhand dieser Matrix entscheiden, welche Kontexte abzulehnen sind und welche angenommen werden können.

Der Anwendungsentwickler kann für das Attribut GRC Schwellwerte des Verwandtschaftsgrades für die zu konsumierenden Kontexte angeben. Dies kann global für alle Kontextpaare geschehen, oder auch in Form einer Matrix, die paarweise Schwellwerte für die jeweiligen Verwandtschaftsgrade enthält.

Bei der Auswertung der Verwandtschaftsmatrix bezüglich der festgelegten Schwellwerte, stellt sich die Frage, welcher der Kontexte eines Kontextpaares, dessen Verwandtschaftsgrad den Schwellwert überschreitet, abgelehnt werden soll. Im Rahmen dieser Arbeit wurden zwei grundlegende Auswahlstrategien implementiert. Im einfachsten Fall wird zufällig einer der beiden Kontexte ausgewählt. Zum Beispiel kann der Kontext, welcher zuerst empfangen wurde akzeptiert werden, während alle später empfangenen Kontexte, die zu diesem einen zu hohen Verwandtschaftsgrad aufweisen, abgelehnt werden.

Eine etwas effektivere Methode überlässt die Auswahl nicht dem Zufall, sondern berechnet für Kontexte, die paarweise den Schwellwert für den Verwandtschaftsgrad überschreiten, einen durchschnittlichen Verwandtschaftswert zu allen anderen Kontexten, die an einer Ableitung oder Fusion beteiligt sind. Der Kontext, der dabei den geringeren mittleren Verwandtschaftsgrad hat, wird akzeptiert, der andere wird verworfen. Im Rahmen der Implementierungen und Simulationen für diese Arbeit wurde auf diese effektivere Methode zur Kontextauswahl zurückgegriffen.

4.10 Zusammenfassung

In diesem Kapitel wurde auf der Basis der Modellierung die Architektur eines Kontextmanagementsystems für ubiquitäre Informationsumgebungen entwickelt. Im Mittelpunkt steht dabei die Ausnutzung von Kontextattributen zur Beschreibung der Kontextqualität. Die Architektur beschreibt den Aufbau und das Zusammenspiel der Komponenten der neu entwickelten Kontextmanagementschicht, um welche die Architektur von Artefakten in ubiquitären Informationsumgebungen erweitert wurde.

Die Architektur konkretisiert die Modellierung der Kontextattribute „Kontextalter (CA)“, „räumliche Herkunft (LOC)“, „Verlässlichkeit (FCR)“ und „Verwandtschaft von Kontexten (GRC)“. Dabei werden die Verfahren erläutert, mit deren Hilfe die Attributwerte der Kontextattribute berechnet und ausgewertet werden.

Für die Umsetzung der Attribute CA und LOC wurden bestehende Verfahren aufgegriffen, erweitert und angepasst, um den Anforderungen der Anwendungsdomäne gerecht zu werden. Das Kontextalter wurde dazu in zwei Parameter, das eigentliche Alter eines Kontextes und seine Ableitungsdauer, aufgespalten. Als Grundlage für die Bestimmung der räumlichen Herkunft von Kontexten dient das RAUM-System. Hier werden *Interessensbereiche* verwendet, um die Anforderungen der Anwendungen an die Herkunft von Kontexten in einer ubiquitären Informationsumgebung zu formulieren und auszuwerten.

Die Algorithmen und Verfahren für die Berechnung und Auswertung der Attribute FCR und GRC wurden im Rahmen dieser Arbeit entwickelt. Sie adressieren konkrete Probleme der Kontextverarbeitung in großen, heterogenen ubiquitären Informationsumgebungen.

FCR nutzt unscharfe Logik, um ein Verlässlichkeitsmaß für Kontexte zu etablieren. Dieses Verlässlichkeitsmaß repräsentiert die Güte des semantischen Kontexterkennungsalgorithmus, durch den der Kontext erzeugt wurde. Semantische Kontexterkennungsalgorithmen ordnen die Eingangsdaten, Sensordaten oder Kontextdaten, Kontextklassen zu und erzeugen basierend auf diesen Eingangsdaten neue Kontextinstanzen dieser Klassen. Das Attribut FCR ist in der Lage, den Abstand einer Kontextinstanz zum Mittelpunkt seiner Kontextklasse zu repräsentieren. Dadurch kann die Güte, beziehungsweise die Verlässlichkeit der Klassenzuordnung, durch einen Konsumenten, evaluiert werden.

GRC führt das Kontextgenom als Basis für die Analyse der informationellen Abhängigkeit von Kontexten ein. Die informationelle Abhängigkeit wird durch den Verwandtschaftsgrad von Kontexten ausgedrückt. Für die Realisierung des Kontextattributes GRC wurden Ideen aus dem Bereich der genetischen Algorithmen und der biologischen Genetik kombiniert, um ein effizientes, Ressourcen schonendes Verfahren für die Bestimmung des Verwandtschaftsgrades zu entwickeln. Mit PMSC wurde ein Crossover-Operator vorgestellt, der den besonderen Anforderungen an die Vererbung von Kontextgenen Rechnung trägt. Darüber hinaus wurde die Stabilität des durch GRC bereitgestellten Verwandtschaftsmaßes durch mathematische Betrachtungen nachgewiesen.

Eine weitere zentrale Komponente der in diesem Kapitel vorgestellten Architektur, ist der Qualitätsfilter. Er hat die Aufgabe die Anforderungen an Kontextattribute, welche von der Anwendung vorgegeben werden, durchzusetzen und eingehende Kontexte entsprechen zu filtern. Dabei können komplexe Vorgaben umgesetzt werden, indem Anforderungen an einzelne Attribute logisch verknüpft werden.

Im nächsten Kapitel wird nun die hier vorgestellte Architektur weiter konkretisiert. Es werden verschiedene Implementierungen, basierend auf realen Artefakten und einer Simulationsplattform eingeführt, welche die Grundlage für die spätere Evaluierung des Qualitätssicherungssystems bilden.

5 Implementierung

Die praktische Umsetzung der, in den vorangehenden Kapiteln entwickelten Methoden zur Qualitätssicherung von Kontexten in ubiquitären Informationsumgebungen, bildet die Grundlage für deren Evaluierung. In diesem Kapitel werden zwei für die Evaluierung des Kontextmanagementsystems relevante Umsetzungen vorgestellt: die konkrete Umsetzung des Systems auf der Basis von Particle Computern zur Evaluierung von realen Anwendungsszenarien im Rahmen der AwareOffice Umgebung und die simulative Umsetzung der Qualitätssicherungsmethoden im Rahmen des „context_sim“ Simulators.

Zunächst wird nun auf die Rahmenbedingungen der Implementierung eingegangen. Die hier vorgestellte Umsetzung ist insgesamt als beispielhaft anzusehen, da mit der Implementierung bestehende Anwendungen aus dem AwareOffice um die in dieser Arbeit entwickelte Kontextmanagementkomponente erweitert wurden.

5.1 Rahmenbedingungen für die Umsetzung des Kontextmanagementsystems

Das in der vorliegenden Arbeit vorgestellte Kontextmanagementsystem hat zum Ziel, die Kontexterkenkung kontextsensitiver Anwendungen in realen ubiquitären Informationsumgebungen zu verbessern. Aus dieser Zielsetzung ergeben sich Anforderungen an die Modellierung und Architektur des Systems, welche in der Entwicklung berücksichtigt wurden. Für die konkrete Umsetzung des Kontextmanagementsystems als Teil einer produktiven ubiquitären Informationsumgebung müssen allerdings weitere Rahmenbedingungen beachtet werden, die von den zur Verfügung stehenden Hardwarekomponenten und der gewählten Testumgebung vorgegeben werden.

Diese Rahmenbedingungen betreffen in erster Linie die Parameterisierung der einzelnen Methoden zur Anpassung an die gewählte Umgebung. Allerdings müssen bei der Implementierung auch Vorgaben der gewählten Hardware beachtet und die Umsetzung entsprechend gestaltet werden.

5.1.1 Reale Anwendungsdomäne

Als Basis für die beispielhafte Umsetzung der in dieser Arbeit vorgestellten Architektur einer ubiquitären Informationsumgebung mit Qualitätssicherungssystem, dient eine Hardwareplattform (die *Particle Computer* [46]), welche entwickelt wurde, um die nachträgliche Computerisierung von Alltagsgegenständen zu ermöglichen und so im Bereich der Ubiquitous Computing Forschung, die schnelle Realisierung ubiquitärer Informationsumgebungen zu ermöglichen.

Particle Computer wurden als Basistechnologie gewählt, da sie zum einen die schnelle Entwicklung von Ubiquitous Computing Anwendungen unterstützen und zum anderen, da Teile der hier vorgestellten Architektur wie AwareCon und ConCom bereits als Referenzimplementierungen verfügbar sind. Particle Computer sind mobile Kleinstrechner, welche auf die *Smart-Its* Plattform [19, 15] zurückgehen. Sie verfügen über einen programmierbaren Mikrokontroller, Speicher, eine Funkschnittstelle zur Kommunikation, sowie verschiedenen Sensoren und Aktuatoren. Für die hier vorgestellte Implementierung kamen *pPart* und *uPart* Particle Computer zum Einsatz.

pPart Particle Computer (*pParts*) verfügen über einen 20MHz Mikrokontroller, 128kByte Programmspeicher, 4kByte Arbeitsspeicher und 512kByte Datenspeicher. Die Funkkommunikation nutzt das lizenzfreie 868MHz Band mit einer Geschwindigkeit von 125kBit/s. Das *pPart* enthält außerdem eine Echtzeituhr und wird über eine Batterie mit Strom versorgt. Als Sensoren stehen ein Beschleunigungssensor (dreiaxial), ein Lichtsensor, ein Temperatursensor, ein Drucksensor und ein einfacher Bewegungssensor (Ballschalter) zur Verfügung. Aktuatoren sind ein Lautsprecher und diverse LEDs (engl.: Light Emitting Diode). Die Größe eines *pParts* beträgt 45x18mm.

uPart Particle Computer (*uParts*) sind weniger leistungsfähig, sie verfügen über einen 4MHz Mikrokontroller und können vom Benutzer nicht programmiert werden. Die *uParts* bieten aber die Möglichkeit der Konfiguration verschiedener Operationsparameter. Mit einer Kantenlänge von nur 17mm, sind sie deutlich kleiner als die *pParts*. Auch die *uParts* werden mit einer Batterie versorgt und kommunizieren im 868MHz Band. Als Sensoren stehen hier ein Lichtsensor, ein Temperatursensor und ein einfacher Bewegungssensor auf der Basis eines Mikroballschalters zur Verfügung. Vom Entwickler ansteuerbare Aktuatoren fehlen.

Die Infrastruktur des Particle Computer Netzwerkes besteht aus Zugangspunkten, den X Bridges. Diese können die Datenpakete der Particle Computer zwi-

schen der Funkschnittstelle und einem angeschlossenen Ethernet umsetzen, indem die Daten dann in UDP-Paketen gekapselt versendet werden.

Für die Entwicklung von Anwendungen auf den pParts stehen eine Reihe von Softwarebibliotheken zur Verfügung, welche den Zugriff auf die Sensorik und Aktuatorik erleichtern. Daneben existiert eine Referenzimplementierung des AwareCon Kommunikationsprotokolls in der Version 5.

Auf der Basis dieser Rahmenbedingungen wurde das hier eingeführte Kontextmanagementsystem als Qualitätssicherungsschicht implementiert, indem die vorhandene Referenzimplementierung des AwareCon Protokolls um die entsprechende Funktionalität erweitert wurde.

5.1.2 Simulation

Die Simulation des Kontextmanagementsystems hat zwei Aufgaben. Zum einen dient sie der Findung geeigneter Parameterisierungen der vorgestellten Qualitätssicherungsmethoden, zum anderen ist die Simulation Grundlage für die Beurteilung der Leistungsfähigkeit der vorgestellten Systeme in Großszenarien, welche die Dimensionen, die in heutigen Testumgebungen untersucht werden können, überschreiten.

Simulationen der hier vorgestellten Methoden zur Qualitätssicherung von Kontexten in ubiquitären Informationsumgebungen wurden in MATLAB und `context_sim` durchgeführt. Die in MATLAB durchgeführten Simulationen dienten vornehmlich der Entwicklung und Verfeinerung der vorgestellten Verfahren, wogegen das Ziel der Simulationen in `context_sim`, zum einen die Bestimmung sinnvoller Systemparameter, zum anderen die Evaluation von großen Anwendungsszenarien, die sich mit den verfügbaren Mitteln nicht real umsetzen ließen, war.

Der Simulator `context_sim` wurde entwickelt, um das systemnahe Kontextverarbeitungsverhalten von Artefakten in einer ubiquitären Informationsumgebung simulieren zu können. Er wurde in JAVA als PC-Software implementiert und stellt den Rahmen für die Simulation dynamischer ubiquitärer Informationsumgebungen zur Analyse der Leistungsfähigkeit von Qualitätssicherungsmechanismen bereit.

Im folgenden Abschnitt wird nun zunächst auf die Simulation von ubiquitären Informationsumgebungen eingegangen und die Entwicklung eines neuen Simulators für diese Umgebungen motiviert. Danach wird die Testumgebung *Aware-*

Office mit ihren kontextsensitiven Anwendungen vorgestellt. Dabei wird gezeigt, wie das Qualitätsmanagementsystem auf den Artefakten dieser Umgebung implementiert wurde.

5.2 Simulation von ubiquitären Informationsumgebungen

Die Simulation von ubiquitären Informationsumgebungen bietet die Möglichkeit Szenarien zu untersuchen, die sich mit realer Hardware heute nicht oder nur sehr schlecht realisieren lassen. Außerdem eröffnet sie die Möglichkeit, schnell verschiedene Konfigurationen von Artefakten und Anwendungen im Zusammenspiel zu analysieren, ohne dass eine Vielzahl von Geräten umprogrammiert werden müssen. Damit ist die Simulation ein wichtiges Werkzeug, das die effiziente Entwicklung zukunftsicher skalierbarer Systeme für ubiquitäre Informationsumgebungen unterstützt [136].

Die Entscheidung, im Rahmen dieser Arbeit einen eigenen Simulator für das Kontextverhaltensverhalten von Artefakten in einer ubiquitären Informationsumgebung zu entwickeln, beruhte auf einer Analyse der verfügbaren Simulatoren im Bereich ubiquitärer Informationstechnologien. Diese hat gezeigt, dass keine Simulatoren verfügbar waren, welche die Anforderungen an die Simulation der Kontextverarbeitung und -kommunikation vieler Artefakte in einer dynamischen ubiquitären Informationsumgebung erfüllen. Bei den verfügbaren Ubicomp-Simulatoren steht die Simulation der Benutzerschnittstellen und der Benutzerinteraktion im Vordergrund. Bei Simulatoren aus angrenzenden Bereichen, wie der Netzwerksimulation oder der Simulation von Agentensystemen, stehen ebenfalls andere Aspekte im Mittelpunkt der Simulation, so dass diese Systeme nur mit erheblichem Aufwand zur Simulation ubiquitärer Informationsumgebungen adaptiert werden können.

5.2.1 Verfügbare Simulatoren

In diesem Abschnitt wird ein Überblick über verfügbare Simulatoren gegeben, die im Rahmen dieser Arbeit auf ihre Tauglichkeit zur Simulation von Kontextkommunikation und Kontextverarbeitung in ubiquitären Informationsumgebungen untersucht wurden. Dabei werden die Eigenschaften, sowie die Ausschlusskriterien der einzelnen Systeme kurz erläutert.

Ubicomp Simulatoren

Unter dem Begriff *Ubicomp Simulatoren* werden hier Simulatoren zusammengefasst, die aus dem Forschungsbereich des Ubiquitous Computing stammen. Der bekannteste dieser Simulatoren ist UbiWise [168, 10, 11] von John Barton und Vikram Vijayaraghavan. UbiWise fokussiert auf die Analyse der Interaktion von Benutzern mit den Artefakten einer ubiquitären Informationsumgebung. Der Simulator basiert auf der 3D-Grafik-Engine des Computerspiels Quake III Arena. In dieser werden die Artefakte und deren Umgebung simuliert. Der Benutzer kann dann mit den virtuellen Gegenständen interagieren und deren Anwendungen ausprobieren.

Eine ähnliche Zielsetzung hat der Simulator TATUS [124, 125] von Eleanor O'Neill. Hier wird allerdings auf die 3D-Grafik-Engine des Computerspiels Half-Life zurückgegriffen. Der Fokus liegt auch hier auf der Erforschung der Interaktion mit Benutzerschnittstellen in Ubiquitous Computing Szenarien.

Beide Simulatoren eignen sich nicht, um Aspekte der systematischen Kontextverarbeitung von Artefakten feinauflösend zu simulieren. Verarbeitungsmodelle für Kontextdaten sind lediglich rudimentär vorhanden. Sie beschränken sich auf die unbedingt notwendigen Funktionen zur visuellen Simulation der Artefakte und ihrer Anwendungen.

Auch der Lancaster Simulator [121] von Ricardo Morla und Nigel Davis und neueste Vorschläge, wie der von Reynolds et al. [136], zielen auf die Unterstützung der Anwendungsentwicklung in Ubiquitous Computing ab. Daher sind auch sie ungeeignet für die Simulation der Funktionalität einer Qualitätssicherungsschicht in ubiquitäre Informationsumgebungen.

Netzwerksimulatoren

Netzwerksimulatoren dienen der Simulation der unteren Schichten des ISO/OSI Basisreferenzmodells. Ihr Fokus liegt auf der Untersuchung physikalischer und protokolltechnischer Einflüsse auf die Kommunikation von zumeist mobilen Geräten. Dabei werden vornehmlich kommunikationsbezogene Parameter wie Durchsatz, Fehlerrate oder Konnektivität betrachtet. Als einer der zahlreichen Vertreter dieser Klasse von Simulationswerkzeugen sei an dieser Stelle der bekannte Netzwerksimulator ns2 [5] genannt.

Die Entscheidung gegen die Verwendung eines Netzwerksimulators als Basis für die Simulation der hier vorgestellten Qualitätssicherungsmechanismen fiel in erster Linie aufgrund der extrem hohen Komplexität dieser Simulatoren und ihrer starken Fokussierung auf Kommunikationsgesichtspunkte. Für die Evaluierung des hier entwickelten Qualitätssicherungssystems ist lediglich das zu Grunde liegende Kommunikationsmodell, wie es in Abschnitt 3.2 identifiziert wurde, von Bedeutung. Alle verarbeitungsbezogenen Komponenten einer Simulationsumgebung hätten auch bei der Verwendung eines Netzwerksimulators von Grund auf neu erstellt und in diesen integriert werden müssen.

Agentenbasierte Simulation

Agentenbasierte Simulatoren wie SWARM [118, 4] und SeSAM [3] wurden entwickelt, um die Interaktion zwischen Agenten und deren Auswirkung auf die Umgebung zu simulieren. Im Zentrum steht dabei die Analyse der Auswirkungen des Verhaltens der Agenten und deren Beziehungen.

Es ist davon auszugehen, dass sich Agenten Simulatoren daher für die Abbildung ubiquitärer Informationsumgebungen eignen. Dabei können einzelne Artefakte als Agenten modelliert werden. Allerdings eignet sich eine solche Simulation nicht zur Untersuchung oder Entwicklung systemnaher Funktionalitäten der Agenten, wie eines Qualitätsmanagementsystems. Vielmehr kann eine solche Simulation für den Test von kollaborativen Anwendungen [47] von Interesse sein.

Fazit

Die Untersuchung verfügbarer Simulationsumgebungen in der Vorbereitung dieser Arbeit hat gezeigt, dass keine dieser Plattformen die Anforderungen an die Simulation eines Qualitätsmanagementsystems für Kontexte in einer ubiquitären Informationsumgebung erfüllt. Im einzelnen waren diese Anforderungen:

- Modellierbarkeit von Artefakten, deren Kommunikation über das CSpace-Modell und deren Qualitätssicherungsschicht.
- Analysierbarkeit der Auswirkungen der Integration der Qualitätssicherungsschicht.
- Einfache Umsetzbarkeit verschiedener Anwendungsszenarien.

Daher wurde im Rahmen dieser Arbeit der Simulator `context_sim` entwickelt, um das systemnahe Kontextverhaltensverhalten von Artefakten in einer ubiquitären Informationsumgebung simulieren zu können. Im folgenden Abschnitt wird der Simulator kurz vorgestellt. Dabei wird vor allem auf die Umsetzung der, in dieser Arbeit entwickelten Verfahren zur Sicherung der QoC eingegangen.

5.2.2 Simulator: `context_sim`

Der Simulator `context_sim` wurde entwickelt, um die Auswirkungen von Qualitätssicherungsmechanismen in großen ubiquitären Informationsumgebungen zu simulieren und diese Mechanismen zu evaluieren. Das Ziel war dabei möglichst genau die Eigenschaften der realen Implementierung umzusetzen, um vergleichbare Ergebnisse aus der Simulation und der Evaluierung in der realen Testumgebung zu erhalten.

Der Simulator ist in der objektorientierten Programmiersprache JAVA als PC-Anwendung geschrieben. In diesem Abschnitt wird die allgemeine Funktionalität des Simulators beschrieben. Die simulatorbasierte Implementierung des QoC Systems wird in Abschnitt 5.2.3 vorgestellt.

Bei der Implementierung von `context_sim` wurde darauf geachtet, dass funktionale Einheiten der ubiquitären Informationsumgebung in eigenen Klassen gekapselt sind. Neben einer Hauptklasse, die Steueraufgaben übernimmt, besteht der Simulator aus einer Artefaktklasse, von der Artefakte verschiedener Typen abgeleitet werden können, einer Kontextklasse, die dazu dient, Kontext-Objekte mit ihren Attributfeldern zu erzeugen, einer Ableitungsklasse, die speichert, wie verschiedene Kontexte in der Simulation von einander abgeleitet werden können und einer Kommunikationsklasse, die einen CSpace implementiert.

`context_sim` simuliert diskrete Zeiteinheiten (Simulationsschritte). In jedem Simulationsschritt werden alle Artefakte einer Umgebung aktiviert und können dann kommunizieren und Kontexte verarbeiten. Der Simulator ist so angelegt, dass die Verarbeitung der Artefakte quasi-parallel stattfindet. Am Ende jedes Simulationsschrittes werden Verwaltungsarbeiten (engl. Maintenance) durchgeführt. Diese beinhalten zum Beispiel die Erstellung zusammenfassender Simulationsstatistiken für jeden Simulationsschritt.

Kommunikation

Die Kommunikation von Artefakten innerhalb der Simulation geschieht über ein CSpace-Objekt. Der CSpace wurde exakt nach den Vorgaben aus Abschnitt 3.7 implementiert. Artefakte können Kontexte in den CSpace schreiben und diese auch wieder aus dem CSpace lesen. Darüber hinaus implementiert der CSpace eine konfigurierbare Kanalzeit, die angibt, für wie viele Simulationsschritte Kontexte, nachdem sie in den CSpace geschrieben wurden, zur Verfügung stehen. Dieser Mechanismus dient dazu verschiedene Kommunikationstopologien zu berücksichtigen. Dazu gehören zum Beispiel segmentierte Netzwerke in denen Infrastrukturkomponenten vorhanden sind, welche Daten replizieren oder puffern können.

Von weitergehenden Eigenschaften der Referenzimplementierung des Aware-Con V5 Protokolls, welches auf der Particle Computer Plattform die Grundlage der Kommunikation bildet, wird in der Implementierung des CSpace abstrahiert.

Kontextsensitive Anwendungen

Anwendungen, die auf den Artefakten einer ubiquitären Informationsumgebung ausgeführt werden, werden im Rahmen von `context_sim` lediglich auf der Basis ihres Kontextverhaltens implementiert. Auf die Simulation von Aktuatorik und Sensorik, inklusive Benutzerschnittstellen, wird vollständig verzichtet. Dafür stellt die Ableitungsklasse zur Laufzeit der Simulation die Informationen darüber bereit, welches Artefakt welche Kontexte konsumieren und welche neuen Kontexte es daraus ableiten kann. Dazu assoziiert die Ableitungsklasse den Typ eines Artefaktes mit den entsprechenden Kontexttypen. Die so entstehenden Ableitungsbäume repräsentieren also einen Kontextverarbeitungsprozess. Ein Artefakttyp kann mit mehreren Ableitungen assoziiert werden.

Kontextverarbeitung

Ist die Qualitätssicherungsschicht nicht aktiviert, folgt die Kontextverarbeitung in `context_sim` einfachen Regeln: Der Typ eines Artefaktes gibt vor, welche Kontextableitungen für dieses Artefakt valide sind. Für jede valide Ableitung wird festgestellt, ob alle benötigten Kontexte im CSpace vorhanden sind. Ist dies der Fall, wird ein neuer Kontext produziert, der dann im nächsten Simulationsschritt

ebenfalls im CSpace verfügbar ist. Sind nicht alle für eine Ableitung notwendigen Kontexte vorhanden, wird der nächste, für das Artefakt gültige Ableitungsbaum abgearbeitet, ohne dass zuvor ein neuer Kontext produziert wurde. Dieser Ablauf setzt sich fort, bis alle für das Artefakt gültigen Ableitungen abgearbeitet sind.

Ableitungsregeln

Neben den Ableitungsbäumen für Kontexte höherer Ordnung (vgl. Abschnitt 4.5), welche die Produktion von Kontexten auf der Basis bereits im CSpace vorhandener Quellkontexte beschreiben, können Ableitungsregeln für die Erzeugung von Kontexten erster Ordnung angegeben werden. Diese Ableitungsregeln zeichnen sich dadurch aus, dass das Feld, welches die Quellkontexte speichert, in diesem Fall leer bleibt.

Als Erweiterung der Funktionalität simulierter Artefakte kann eine Übergangswahrscheinlichkeit zwischen alternativ zu verwendenden Ableitungsregeln angegeben werden.

5.2.3 Qualitätsmanagementschicht: QoC

Die Funktionalität der Qualitätsmanagementschicht ist in `context_sim` als Teil der Artefaktklasse implementiert. Kontextattribute werden von den Kontextobjekten als Objektvariablen zur Verfügung gestellt. Diese werden bei der Erzeugung des Kontextes durch den Qualitätsaggregator des erzeugenden Artefaktes gesetzt.

Neben den Algorithmen zur Auswertung und Berechnung der Kontextattribute, stellt der Qualitätsfilter die zentrale Komponente der Qualitätsmanagementschicht der Artefakte dar. Dieser ist in `context_sim` so implementiert, dass sich die Funktionalität einzelner Kontextattribute für die Simulation zu- beziehungsweise abschalten lässt. Dadurch ist es möglich, Vergleiche zwischen der Kontexterkennungslleistung der Artefakte mit und ohne Qualitätsmanagement in einem Anwendungsszenario zu ziehen. Die simulative Evaluierung geschieht dann, wie in Kapitel 6 gezeigt wird, mittels verschiedener Simulationsläufe gleicher Anwendungsszenarien, bei denen die Parameter der Kontextattribute verändert werden.

Simulation von Anwendungsszenarien

Die Simulation von Anwendungsszenarien mit `context_sim` basiert auf der Art und Anzahl der simulierten Artefakte und deren Typen, die für die Simulation mittels der Ableitungsregeln, deren Kontextverhaltensverhalten vorgeben. Daneben lassen sich, wie bereits erwähnt, die Parameter der Kontextqualitätssicherung konfigurieren. Simulierte Szenarien werden außerdem über den, für das Szenario vorgegebenen RAUM-Lokationsbaum beeinflusst. Weitere Parameter der Simulation sind das Aktivitätsniveau der einzelnen Artefakte, welches angibt, wie oft ein Artefakt im Durchschnitt in der Simulation aktiviert wird und die *echte Zuverlässigkeit* der produzierten Kontexte.

Da der Simulator nicht die kontextsensitive Anwendung selbst simuliert, sondern nur deren Verhaltenseverhalten, muss ein Wert für die tatsächliche Zuverlässigkeit der simulierten Kontexte angegeben werden. Anhand dieses Wertes wird entschieden, wie viele der simulierten Kontexte eines Artefaktes korrekt sind. Sie entspricht also der Kontexterkennungsrate des Artefaktes und dient als Basis für die Evaluierung des Qualitätsmanagementsystems, das zum Ziel hat, diese Kontexterkennungsrate bei Artefakten, die Kontexte höherer Ordnung verarbeiten, zu verbessern (vgl. These dieser Arbeit).

5.3 Reale Artefakte: AwareOffice

Grundlage für die Evaluierung des Qualitätsmanagementsystems im realen Anwendungsumfeld, ist das AwareOffice und seine Artefakte. Das AwareOffice dient als Test- und Evaluierungsumgebung für Ubiquitous Computing Anwendungen und Geräte. Dabei folgt es dem Prinzip des lebenden Labors (engl. Living Lab) [6, 97]. Das bedeutet, die Testumgebung wird im täglichen Lehr- und Forschungsbetrieb benutzt und dabei einer kontinuierlichen Evaluierung unterzogen.

Das AwareOffice ist eine Büroumgebung, die ihre Benutzer bei typischen Aufgaben im Büroumfeld aktiv unterstützen kann. Artefakte in dieser Umgebung umfassen unter anderem Stühle, Tassen und Boardmarker zum Schreiben auf einer Weißwandtafel und den dazu gehörenden Schwamm. Hinzu kommen PDAs, ein interaktives Türschild, eine Digitalkamera und ein Datenprojektor, sowie kontextsensitive Fenster und eine Tür.

In den folgenden Abschnitten werden diese Artefakte und die unterschiedlichen Implementierungen der Qualitätsmanagementschicht auf diesen vorgestellt. Eine

ausführliche Beschreibung der kontextsensitiven Anwendungen der Artefakte des AwareOffice findet sich in [190].

5.3.1 AwarePen

Der AwarePen ist ein Stift zum Schreiben auf einer Weißwandtafel, welcher in der Lage ist, die Art seiner Benutzung zu erkennen. Es existieren insgesamt drei verschiedene Implementierungen des AwarePens, die sich durch die zur Kontexterkenennung verwendeten Algorithmen unterscheiden: die erste hier vorgestellte Implementierung basiert auf der Anwendung von Filterketten. Eine weitere Implementierung nutzt neuronale Netzwerke zur Kontexterkenennung. Die dritte Implementierung basiert auf unscharfer Logik. Aufgrund dieser Diversifizierung ist es möglich, in der Evaluierung des Kontextmanagementsystems den Einfluss verschiedener Algorithmen auf die erreichbaren Erkennungsraten zu berücksichtigen, auch wenn die betrachtete Anwendung unverändert bleibt.

Alle drei Implementierungen sind in der Lage, die gleichen drei Kontexte des AwarePen zu unterscheiden und diese zu kommunizieren. Diese Kontexte können als „stillliegen“, „spielen oder gestikulieren“ und „schreiben“ beschrieben werden.

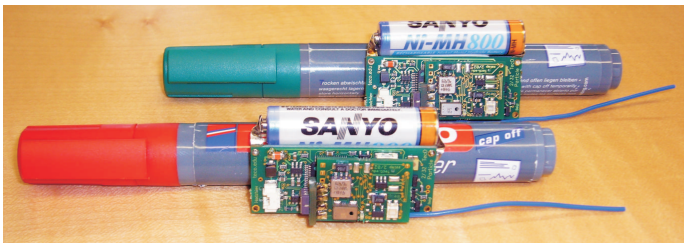


Abbildung 5.1. AwarePens: Tafelstifte mit pParts

Die Hardware des AwarePen besteht, wie Abbildung 5.1 zeigt, aus einem handelsüblichen Weißwandtafelstift, an dem ein Particle Computer von Typ pPart befestigt wurde. Als Sensorboard kommt ein Ssimp mit einem dreiachsigen Beschleunigungssensor zum Einsatz.

Implementierung der Qualitätsmanagementschicht auf pParts

Auf den, auf pPart Particle Computern basierenden Artefakten, ist das Qualitätsmanagementsystem als Erweiterung des AwareCon V5 Kommunikationsprotokolles [17] implementiert. Dazu wurde der Protokollkopf der ACL-Schicht (engl. Application Convergence Layer) um Felder zur Aufnahme der Kontextattribute erweitert und die Kontextmanagementschicht mit dem Qualitätsfilter und Qualitätsaggregator als Erweiterung der Funktionalität des Protokollstapels implementiert. Abbildung 5.2 zeigt den Aufbau des erweiterten Protokollstapels.

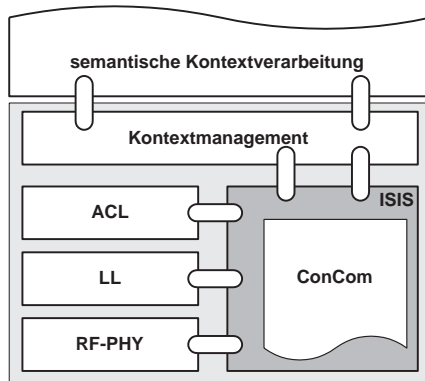


Abbildung 5.2. AwareCon mit Kontextmanagementschicht

Der erweiterte ACL-Kopf ist in Abbildung 5.3 dargestellt. Das Feld *ACL-Typ* enthält den 3-2-kodierten¹ Kontexttyp in zwei Bytes. Danach folgt das ein Byte lange Feld *DL*. In ihm ist die Länge der in diesem Paket enthaltenen semantischen Kontextdaten in Byte abgelegt.

Die Felder *CSA* und *CPA* enthalten die Zeitstempel des Kontextattributes *CA*. Beide sind in je 4 Bytes kodiert. Dabei wurde für die Evaluierung eine einfache reduzierte Darstellung verwendet, welche lediglich ein Byte für die Minuten, ein Byte für die Sekunden und zwei Bytes für die Millisekunden vorsieht. Diese Reduktion ist möglich, da es sich bei AwareCon V5 um ein zeitsynchronisiertes

¹Die 3-2-Kodierung [14] erlaubt das Speichern von drei Zeichen eines reduzierten Alphabets in 2 Datenbytes.

Protokoll handelt. Die Zeitreferenz wird im Netzwerk von Zeitservern bereitgestellt.

ACL-Typ	DL	CSA	CPA	LOC	FCR	GRC
2 Bytes	1	4	4	2	2	30

Abbildung 5.3. Erweiterter ACL-Protokollkopf

Darauf folgen zwei Bytes, welche das Attribut *LOC* enthalten. Auch hier wird eine reduzierte Darstellung verwendet. Die zwei Bytes enthalten eine eindeutige Referenz in den RAUM-konformen Lokationsbaum der ubiquitären Informationsumgebung. Diese Darstellung wurde gewählt, da der Protokollkopf der Verbindungsschicht (engl. Link Layer, LL) des AwareCon-Stapels bei der Kommunikation über drahtgebundene Netzwerke bereits die vollständige Lokation eines Artefaktes in RAUM-konformer Kodierung enthält.

Das Feld *FCR* enthält die Verlässlichkeit des Kontextes. Der zwei Bytes umfassende Wert wird intern als Gleitkommazahl im Wertebereich zwischen 0 und 1 verarbeitet. Abschließend enthält das Feld *GRC* das Genom des Kontextes hierfür sind 30 Bytes reserviert. Die Aufteilung dieses Feldes richtet sich nach der Konfiguration des Kontextgenoms. Die Anzahl der Gene und ihrer Allele ist innerhalb der verfügbaren 30 Bytes frei konfigurierbar.

5.3.2 AwareSponge

Der AwareSponge (Abbildung 5.4) ist ein Schwamm für das Wischen von Weißwandtafeln, der mit einem pPart Particle Computer ausgestattet ist. Das hier verwendete Sensorboard enthält Bewegungs- und Kraftsensoren. Mit diesen kann erkannt werden, wann der Schwamm dazu benutzt wird, die Tafel zu wischen. Die Kontexterkenkung des AwareSponge basiert auf einer Filterkette. Sie unterscheidet die Zustände „in Ruhe“ und „es wird gewischt“.

Die Implementierung des Kontextmanagementsystems entspricht der des AwarePen, da hier ebenfalls pParts als Basistechnologie des Artefaktes eingesetzt werden.

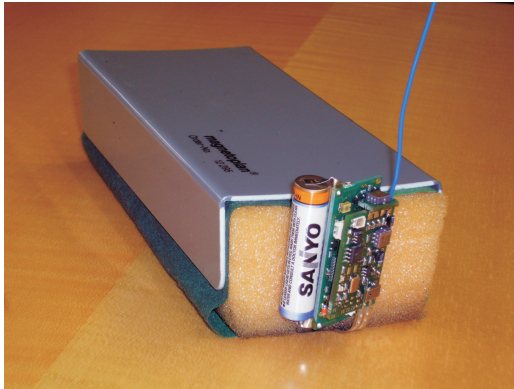


Abbildung 5.4. AwareSponge: kontextsensitiver Tafelschwamm

5.3.3 AwareCam

Die AwareCam ist ein Artefakt, welches aus einer, von einem PC gesteuerten Digitalkamera besteht. Die Aufgabe der Kamera ist es, während einer Besprechung, Bilder von den Tafelaufschrieben an der Weißwandtafel während einer Besprechung zu machen und diese per Email an die Teilnehmer zu versenden.

Die Anwendung arbeitet dabei auf der Basis verschiedener Kontexte, die im AwareOffice verfügbar sind, so zum Beispiel den Kontexten, welche von den AwarePens und dem AwareSponge bereitgestellt werden. Hinzu kommt der „Besprechungs“-Kontext, der von dem, in Abschnitt 5.3.5 vorgestellten, interaktiven Türschild erzeugt wird.

Die AwareCam Software wird auf einem PC ausgeführt, an den eine Digitalkamera angeschlossen ist. Die Kontextkommunikation ist auch hier als Erweiterung des AwareCon Protokolls implementiert. Dieses steht neben der Version für Particle Computer auch in einer PC Version (libParticle²) zur Verfügung. Um Kompatibilität zwischen den verschiedenen Hardwareplattformen zu gewährleisten, entspricht die Implementierung der Kontextmanagementschicht der, auf den Particle Computern.

²libParticle: <http://particle.teco.edu/software/libparticle/index.html> (besucht 10/2006)

Die AwareCam kann feststellen wie viel Text auf die Tafel geschrieben wurde und erkennt, ob sich eine Person, zum Beispiel der Schreiber, im Bildausschnitt befindet. In Schreibpausen versucht die Anwendung dann ein Foto der Tafel aufzunehmen. Greift der Schreiber zum Schwamm bevor die Kamera eine Gelegenheit hatte ein Bild aufzunehmen, kann dieser gewarnt werden. Eine ausführliche Beschreibung der AwareCam Anwendung und der zum Einsatz kommenden Bilderkennung, ist in [112] zu finden.

5.3.4 Stühle, Fenster, Tür, Projektor, Weißwandtafel

Die Artefakte, welche in diesem Abschnitt zusammengefasst werden basieren auf der uPart Hardwareplattform. Wie bereits in Abschnitt 5.1 erläutert wurde, handelt es sich bei den uPart Particle Computern um Geräte, die nicht vom Benutzer programmiert, sondern lediglich durch Konfiguration den Anforderungen einer Ubiquitous Computing Anwendung angepasst werden können. Dadurch ist es nicht möglich eine Qualitätsmanagementschicht direkt auf den uParts zu implementieren. Da sich das Paketformat der drahtlosen Kommunikation von uParts vom dem, der pParts (dem Standard AwareCon V5 Format) unterscheidet, muss dieses von Infrastrukturobjekten im AwareOffice umgesetzt werden.



Abbildung 5.5. Stuhl mit uPart



Abbildung 5.6. Projektor mit uPart

Auf der Basis dieser Umsetzung wurde im Rahmen einer Diplomarbeit [109] das Konzept der virtuellen pParts entwickelt. Dabei handelt es sich um einen Proxyserver, der uPart Pakete entgegen nimmt und diese in gültige pPart umwandelt, bevor sie wieder in das Kommunikationsnetzwerk gesendet werden. Pakete,

die von den virtuellen pParts (den vParts) gesendet werden, entsprechen im Aufbau dem Paketformat aus Abbildung 5.3.

Neben der Konvertierung des Paketformates, ist es die Aufgabe des vPart Proxyserver, die kontextsensitiven Anwendungen der verschiedenen uPart basierten Artefakte zu implementieren und die Funktionalität der Kontextmanagementschicht bereitzustellen. Da der vPart Proxyserver auf der libParticle aufsetzt, wurde hier auf dieselbe Implementierung des Kontextmanagements zurückgegriffen, welche auch die Basis des AwareCam Servers bildet.



Abbildung 5.7. Fenster mit uPart



Abbildung 5.8. Tür mit uPart

Die Kontexte, welche von den hier vorgestellten Artefakten produziert werden, dienen unter anderem dazu, Besprechungen im AwareOffice zu detektieren. Die kontextsensitiven Stühle (Abbildung 5.5) können unterscheiden, ob sie besetzt oder frei sind. Der Datenprojektor (Abbildung 5.6) produziert die Kontexte „in Betrieb“ oder „ausgeschaltet“. An der Weißwandtafel ist ein uPart angebracht, über welches festgestellt werden kann, ob die Tafel benutzt wird. Fenster und Türen (Abbildungen 5.7 und 5.8) können die Kontexte „geöffnet“ und „geschlossen“ kommunizieren.

5.3.5 Interaktives Türschild

Das interaktive Türschild (Abbildung 5.9) stellt, aus Sicht der Kontextverarbeitung, die zentrale Komponente des AwareOffice dar. Es führt die Mehrzahl der im AwareOffice verfügbaren Kontexte zusammen und produziert selbst einen Besprechungskontext. Darüber hinaus stellt das Türschild ein Interface zu einem Besprechungsraumbuchungssystem bereit. Dieses Buchungssystem kann genutzt werden, um den Besprechungsraum zu reservieren. Finden spontane,

nicht vorab geplante Besprechungen statt, die von den Artefakten des Aware-Office erkannt werden, sorgt das Türschild dafür, dass auch diese Belegungen automatisch in das Buchungssystem eingetragen werden.

Das interaktive Türschild zeigt außen, neben dem Besprechungsraum, ständig den Zustand des Raumes an. Mögliche Kontexte sind hier „eine Besprechung findet statt“, „es herrscht leichte Aktivität im Besprechungsraum“ und „der Besprechungsraum ist frei“.

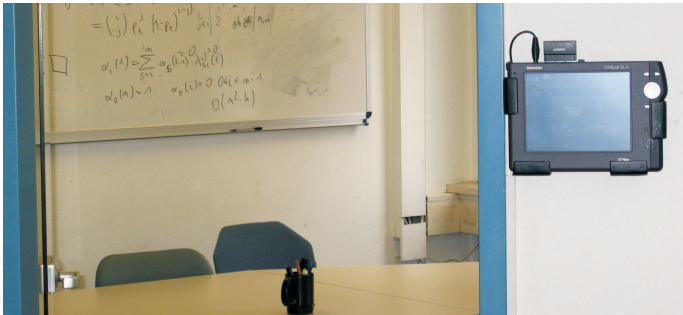


Abbildung 5.9. Interaktives Türschild am AwareOffice

Die Hardware des Türschildes besteht aus einem Siemens SIMPad SL4, einem Webpad Computer mit einem 8,5 Zoll großen berührungssensitiven TFT Bildschirm. Die Netzwerkverbindung ist über WLAN realisiert. Die Anwendung des Türschildes ist PC basiert und baut auf der bereits vorgestellten Implementierung auf der Basis der libParticle auf.

5.3.6 Kontextsensitive PDAs

Die kontextsensitiven PDAs sind Standard PDAs, die zusätzlich mit einem pPart Particle Computer ausgerüstet sind. Diese PDAs wurden ursprünglich im Rahmen des MemoClip II Projektes [44, 186] entwickelt. Sie werden, mit erweiterter Funktionalität, im AwareOffice eingesetzt.

Die PDAs sind in der Lage, unabhängig vom interaktiven Türschild, einen Besprechungskontext zu produzieren. Alternativ können sie aber auch den Besprechungskontext des Türschildes konsumieren. Diese Kontexte werden auf den

PDA's dazu verwendet, die Erinnerungsfunktion dynamisch der umgebenden Situation anzupassen. Kalendererinnerungen werden dabei während einer Besprechung nur optisch angezeigt, eine akustische Alarmierung wird unterbunden.

Die pPart Particle Computer der PDA's enthalten dieselbe AwareCon-Erweiterung, wie sie auch in den AwarePens zum Einsatz kommt.

5.4 Zusammenfassung

Dieses Kapitel stellt die der Evaluierung zu Grunde liegende Implementierung, des in dieser Arbeit entwickelten Kontextmanagementsystems vor. Dabei werden die Bereiche *Simulation* und *Anwendung in einer realen ubiquitären Informationsumgebung* unterschieden.

Für die Simulation der Auswirkungen eines Qualitätsmanagementsystems wurde im Rahmen dieser Arbeit der Simulator `context_sim` entwickelt und implementiert. Dieser Simulator ist darauf spezialisiert, die Kommunikation und Kontextverarbeitung von Artefakten in ubiquitären Informationsumgebungen gemäß der Modelle, welche dieser Arbeit zu Grunde liegen zu simulieren. Dadurch wird es möglich, Middleware, wie das Kontextmanagementsystem, simulativ zu evaluieren. Dabei sind vor allem Szenarien von Interesse, die eine Größe und Komplexität erreichen, welche mit dem heutigen Stand der Technik nicht in realen Testumgebungen umgesetzt und analysiert werden können. Der Simulator stellte außerdem während der Entwicklung des Kontextmanagementsystems ein wichtiges Werkzeug dar, um Anpassungen schnell und effizient testen zu können.

Parallel zu der Umsetzung des Kontextmanagementsystems in einer Simulatore Umgebung, steht die Implementierung des Systems auf realer Hardware. Dafür wurden die Artefakte der AwareOffice Umgebung ausgewählt, da diese Umgebung dem Prinzip des Living Lab folgt und beste Voraussetzungen für die realistische Evaluierung des Qualitätssicherungssystems bietet. Durch die Implementierung des Qualitätsmanagementsystems auf der Particle Computer Hardware konnte der Nachweis geführt werden, dass das vorgestellte System in der Praxis auf Ubiquitous Computing Plattformen implementierbar ist.

Neben der javabasierten Implementierung der Kontextmanagementschicht im Simulator `context_sim`, wurden in diesem Kapitel zwei weitere Implementierungen vorgestellt. Beide setzen das Qualitätssicherungssystem in der realen Anwen-

domäne um. Sie wurden in das Particle Computer System integriert, welches die Grundlage für die Anwendungen im AwareOffice bildet.

Die erste dieser Umsetzungen läuft direkt auf der Particle Computer Hardwareplattform, die zweite wurde in die PC basierte Implementierung des Particle Computer Kommunikationssystems integriert. Somit sind nun alle aktiven Komponenten des AwareOffice mit dem, in dieser Arbeit vorgestellten Kontextmanagementsystem zur Sicherung der Qualität von Kontexten in ubiquitären Informationsumgebungen ausgestattet.

Ziel des hier entworfenen und umgesetzten Systems ist die Verbesserung der Kontexterkenkung von Artefakten in ubiquitären Informationsumgebungen. Im folgenden Kapitel wird nun auf der Basis der hier vorgestellten Implementierungen, das Kontextmanagementsystem, im Hinblick auf dieses Ziel, evaluiert.

6 Evaluierung

In diesem Kapitel werden die Ergebnisse der Evaluierung, des in dieser Arbeit entwickelten Kontextmanagementsystems zur Sicherung der Qualität von Kontextdaten in ubiquitären Informationsumgebungen, vorgestellt. Mit den Ergebnissen aus simulierten und realen Anwendungsumgebungen werden die Thesen der Arbeit gestützt und die Effektivität des Systems nachgewiesen. Die Evaluierung ist in drei Abschnitte unterteilt die verschiedene Systemaspekte näher beleuchten und so ein umfassendes Bild der Leistungsfähigkeit des Systems zeichnen. Dazu werden die einzelnen Kontextattribute und ihre Auswirkung auf die Kontexterkennungsrate von Artefakten in ubiquitären Informationsumgebungen untersucht.

Zuerst werden generelle Eigenschaften des Attributes GRC anhand von Simulationen in MATLAB dargestellt. Dieses Attribut und seine gesamte algorithmische Umsetzung wurden im Rahmen dieser Arbeit entwickelt. Dabei wurde die mathematische Software und Programmiersprache MATLAB als zentrales Werkzeug eingesetzt.

Der zentrale Teil dieses Kapitels enthält die Ergebnisse der simulativen Evaluierung des Kontextmanagementsystems. Dabei wurde der Simulator `context_sim` eingesetzt, um die Auswirkungen der einzelnen Attribute auf die Kontexterkennungsrate von Artefakten zu analysieren. Hierzu wurden typische Problemszenarien betrachtet. Abschließend wurde die Leistungsfähigkeit des Gesamtsystems und das Zusammenspiel aller Attribut in einer integrierten Simulation beurteilt.

Der dritte Teil der Evaluierung fand in der realen Anwendungsumgebung des AwareOffice statt. Hier wird anhand einer ausgewählten Anwendung, dem AwarePen, die Verbindung zu den simulativen Ergebnissen hergestellt und diese mit den im AwareOffice erzielten Ergebnissen verglichen.

6.1 Simulation: GRC in MATLAB

Die mathematische Programmier- und Analyseumgebung MATLAB diente während der Entwicklung des Kontextattributes GRC als zentrales Werkzeug. In die-

sem Abschnitt werden spezielle Eigenschaften dieses Attributes analytisch betrachtet. Dazu werden Daten aus der Entwicklungsphase herangezogen. Den Untersuchungen liegt eine algorithmische Implementierung von GRC in MATLAB zu Grunde. Das zentrale Interesse gilt bei den hier vorgestellten Untersuchungen den Auswirkungen der Parameterauswahl für das Attribut GRC.

6.1.1 Adjustierbare Parameter von GRC

Die frei wählbaren Parameter, welche zur individuellen Anpassung des Attributes GRC an eine ubiquitäre Informationsumgebung genutzt werden können, sind die Anzahl der Gene n , aus denen ein Kontextgenom besteht und die Anzahl der Allele r , die jedes Gen des Genoms kodieren kann. Werden, wie hier Bit-Vektoren zur Repräsentation des Genoms eingesetzt, stellt die Formel $l = \frac{n \ln r}{\ln 2}$ (Formel 4.22) den Zusammenhang zwischen der Länge des gesamten Kontextgenoms l und den Parametern n und r her.

Das Auflösungsvermögen des Verwandtschaftsgrades hängt von n ab. Der Verwandtschaftsgrad wird durch eine reelle Zahl aus dem Intervall $[0, 1]$ repräsentiert. Die Anzahl der möglichen Werte, welche der Verwandtschaftsgrad annehmen kann ist damit $n + 1$ (vgl. Formeln 4.23 und 4.24).

Der Parameter r des Kontextgenoms beeinflusst den systematischen Fehler des Verfahrens. Dies ergibt sich aus der Formel für die Obergrenze des zu erwartenden Fehlers $E(rel_{err}) = \frac{1}{m} \sum_{i=1}^m \frac{i-1}{r}$ (vgl. Formel 4.30). Die Variable m bezeichnet hier die Anzahl der an einem Vererbungsprozess beteiligten Elterngenome.

6.1.2 Auswirkung der Parameter Variation

In diesem Abschnitt werden die Auswirkungen der Variation der Parameter n und r auf das Attribut GRC untersucht. Dazu wird zunächst eine Versuchsreihe mit variabler Anzahl an Genen im Genom betrachtet. Für diese Simulationen wurde $r = 256$ gewählt und auf diesem Wert festgehalten.

Beispielhaft dient eine einfache Vererbung als Basis der Versuche. Aus zwei Kontext Genen C_0 und C_1 wird ein Kindgenom C_2 erzeugt. Beim Crossover zwischen C_0 und C_1 ist die Vererbungswahrscheinlichkeit von Genen beider Genome gleich 0,5. Betrachtet wird dann die Verteilung der Rückgabewerte von

$rel(C_1, C_2)$, des Verwandtschaftsgrades des Kindgenoms und eines der Elterngenome. Hierbei bildet die Varianz $Var(rel(C_1, C_2))$ ein Maß für die Güte des durch GRC berechneten Verwandtschaftsgrades.

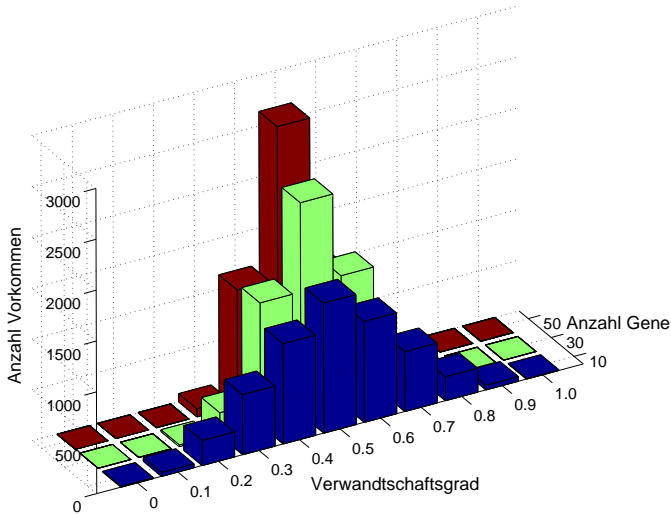


Abbildung 6.1. Verteilung der Verwandtschaftsgrade für $n = 10, 30, 50$

Abbildung 6.1 zeigt die Versuche für $n = 10$, $n = 30$ und $n = 50$. Bei jedem der Versuche wurden die resultierenden Verwandtschaftsgrade aus 5000 unabhängigen Vererbungsprozessen ausgewertet. Die Klassenbreite des Histogramms beträgt hier 0,1. Es ist zu erkennen, dass die Varianz der Verteilung der Verwandtschaftsgrade mit steigendem n abnimmt. Die Abbildung 6.2 zeigt die Versuche $n = 100$, $n = 500$ und $n = 1000$ mit einer feineren Auflösung. Hier wurde als Klassenbreite des Histogramms 0,01 gewählt.

Formal stellt sich dieser Sachverhalt folgendermaßen dar: Für den Erwartungswert der Standardabweichung des Verlässlichkeitsmaßes rel gilt:

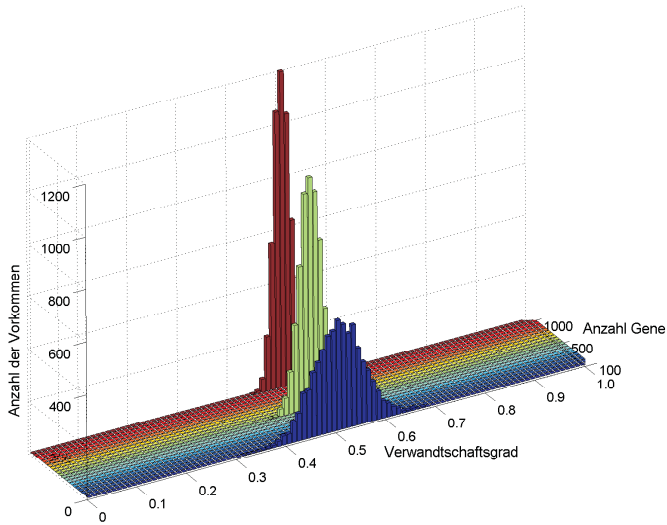


Abbildung 6.2. Verteilung der Verwandtschaftsgrade für $n = 100$, $n = 500$ und $n = 1000$

$$E(\sigma(rel)) = \frac{\sqrt{n \cdot rel \cdot (1 - rel)}}{n} \quad (6.1)$$

Daraus ergibt sich für den Erwartungswert der Varianz von rel :

$$E(Var(rel)) = \left(\frac{\sqrt{n \cdot rel \cdot (1 - rel)}}{n} \right)^2 = \frac{n \cdot rel \cdot (1 - rel)}{n^2} = \frac{rel \cdot (1 - rel)}{n} \quad (6.2)$$

Da der Term $(rel \cdot (1 - rel))$ für $rel = 0,5$ maximal wird, zeigen die obigen Versuche die größte zu erwartende Varianz in den Verwandtschaftsgraden bei gegebenem n . Ein Wert von $rel = 0,5$ stellt also den schlechtesten Fall für die Ver-

teilung der Verwandtschaftsgrade dar. Die Tabelle 6.1 zeigt die Ergebnisse der Versuche noch einmal im Überblick.

n	$\sigma(\text{rel}(C_1, C_2))$	$\text{Var}(\text{rel}(C_1, C_2))$	$\overline{\text{rel}}(C_1, C_2)$
10	~0,159	0,025173	0,4998
30	~0,09	0,0081651	0,49979
50	~0,07	0,004905	0,4991
100	~0,049	0,002413	0,50098
500	~0,022	0,00049966	0,50139
1000	~0,016	0,0002479	0,50093

Tabelle 6.1. Auflösungsvermögen von GRC

Für die weiteren Versuche im Rahmen dieser Evaluierung wurde $n = 30$ gewählt, da dies den Beschränkungen der Paketlänge in der Implementierung auf den pParts entspricht. Auch für die simulativen Versuche wurde diese Anzahl Gene gewählt, um sicher zu stellen, dass die Ergebnisse der Simulationen mit denen der Feldversuche direkt vergleichbar sind.

Der zweite variable Parameter von GRC ist die Anzahl der Allele r . Die Auswirkung der Anzahl der Elterngenome, die an einem Vererbungsprozess beteiligt sind, wurde bereits in Abschnitt 4.8.4 untersucht. Hier wird nun die Auswirkung der Variation der Anzahl der Allele auf den zu erwartenden Fehler in der Berechnung des Verwandtschaftsgrades $E(\text{rel}_{err}) = \frac{1}{m} \sum_{i=1}^m \frac{i-1}{r}$ analysiert.

Für die folgende Betrachtung wird die Anzahl der Elterngenome, welche an einem Verarbeitungsprozess beteiligt sind auf 2 festgesetzt. Die Formel für den zu erwartenden Fehler ist in diesem Fall $E(\text{rel}_{err}) = \frac{1}{2r}$. Abbildung 6.3 zeigt die Entwicklung des Fehlers bei einer steigenden Anzahl Allele. Zu beachten ist, dass hier für die X-Achse eine exponentielle Darstellung gewählt wurde. Bei 256 Allelen, also einer Genlänge von 8 Bit, liegt der zu erwartende Fehler bei bis zu 6 Elterngenomen unterhalb eines Wertes von 0,01. Daher wurde diese Einstellung für die weitere Evaluierung gewählt.

6.2 Simulation: context_sim

In diesem Abschnitt werden die Ergebnisse der simulativen Evaluierung des Kontextmanagementsystems vorgestellt. Als Bewertungskriterium für die Evaluie-

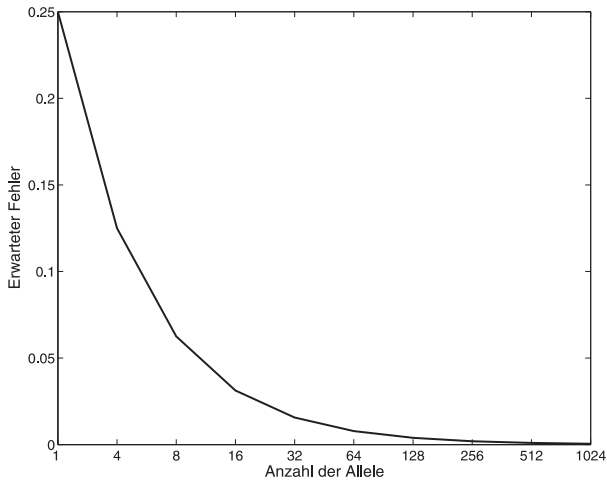


Abbildung 6.3. Erwarteter Fehler in Abhängigkeit von der Anzahl der Allele

rung dient die Kontexterkennerungsrate einer kontextsensitiven Anwendung. Um die Auswirkung der Nutzung von Kontextattributen durch das Kontextmanagementsystem zu quantifizieren, werden die im Folgenden beschriebenen Versuche einmal ohne und zum Vergleich einmal mit aktiviertem Kontextmanagementsystem durchgeführt.

Die Problematik bei der quantitativen Beurteilung der Verbesserung der Kontexterkennerungsrate, liegt in der Komplexität ubiquitärer Informationsumgebungen. Durch die komplexen Abhängigkeiten von Anwendungen und die hohe Dynamik in den Umgebungen ist es kaum möglich, Verbesserungen der Kontexterkennerungsrate eindeutig der Anwendung eines bestimmten Attributes zuzuordnen.

Die mittlere Kontexterkennerungsrate aller Artefakte einer Umgebung ist damit wenig aussagekräftig. Dies ist auf verschiedene Faktoren zurückzuführen: Die Kontextattribute wurden entwickelt, um konkrete in ubiquitären Informationsumgebungen typische Problemsituationen behandeln zu können. Die Verteilung und Gewichtung dieser Probleme innerhalb einer ubiquitären Informationsumgebung kann aber weder in realen Versuchen, noch in realitätsnahen Simulationen so sta-

bil gehalten werden, dass sinnvolle quantitative Analysen der Attributauswirkungen auf die globale Erkennungsrate möglich werden.

Daher wird hier ein anderer Ansatz gewählt: Zunächst wird für jedes der Kontextattribute ein Versuchsaufbau definiert, welcher die Problemsituation, die das Attribut adressiert, beinhaltet. In dieser Versuchsanordnung wird dann die Erkennungsrate eines oder weniger bestimmter Artefakte beobachtet. Dabei wird der Versuch einmal ohne aktiviertes Kontextmanagement durchgeführt. Die Ergebnisse werden dann verglichen mit einem Versuchslauf, in dem das zu untersuchende Attribut aktiviert ist. Somit kann die Auswirkung jedes einzelnen Attributes auf die Kontexterkennungsrate eines Artefaktes in einer speziellen Problemsituation evaluiert werden. Überschneidungen der Effekte von Kontextattributen werden dabei vermieden. Die Versuchsanordnungen bleiben überschaubar und nachvollziehbar.

Abschließend wird anhand eines integrierten Gesamtszenarios die Auswirkung des vollständigen Kontextmanagementsystems auf die globale Kontexterkennungsrate einer Versuchsumgebung betrachtet. Dabei werden die zuvor untersuchten Versuchsanordnungen zusammengefügt und untereinander verbunden.

6.2.1 Attribut CA: Kontextalter

Das Kontextattribut CA beschreibt das Alter eines Kontextes. Fehler in der Kontexterkennung können aufgrund des Kontextalters entstehen, wenn die Gleichzeitigkeit von Ereignissen eine Rolle für die richtige Beurteilung der Situation spielt. Ebenso kann die zeitliche Abfolge von Eingangskontexten eine Rolle für die korrekte Interpretation einer Situation spielen. Hier werden nun zwei Versuche vorgestellt, welche den Nutzen des Attributes CA in Bezug auf das Produktionsalter (CPA) und das Keimalter (CSA) von Kontexten, analysieren.

Versuchsaufbau: CPA

Ein Anwendungsbeispiel für die Relevanz der Gleichzeitigkeit von Ereignissen stammt aus dem AwareOffice. Bei der Erkennung einer Besprechungssituation werden unter anderem die Kontexte der Stühle herangezogen. Sind mehrere Stühle aktiv, ist davon auszugehen, dass sich mehrere Personen im Raum befinden. Die Wahrscheinlichkeit für eine Besprechung steigt. Mehrere Stühle können aber

auch während der Reinigung, beim Saugen des Raumes, aktiviert werden, wenn diese nacheinander vom Tisch weggezogen und wieder zurückgeschoben werden. Ausschlaggebend ist hier die Zeitspanne, innerhalb der die Kontexte der Stühle entstehen.

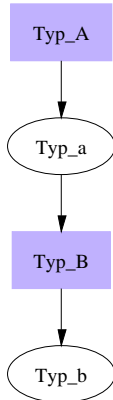


Abbildung 6.4. Simulation CPA:
Subskriptionsgraph

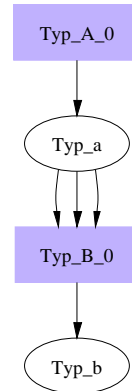


Abbildung 6.5. Simulation CPA:
Ableitungsgraph

Die Simulation dieses Beispiels besteht aus 11 Artefakten. 10 Artefakte des Typs A, welche einen Kontext von Typ a produzieren, entsprechen den Stühlen im Anwendungsbeispiel. Ein Artefakt vom Typ B aggregiert die Kontexte vom Typ a zu einem Kontext vom Typ b. Dieser entspricht dem Besprechungskontext. Abbildung 6.4 zeigt den Subskriptionsgraph. Im Ableitungsgraph in Abbildung 6.5 ist zu erkennen, dass das Artefakt B mindestens drei Kontexte des Typs a benötigt, um einen Kontext vom Typ b zu produzieren. Bezogen auf das obige Beispiel würde dies bedeuten, dass eine Besprechung erkannt werden soll, wenn mindestens drei Stühle aktiv sind.

Die Kontextalgorithmen aller Artefakte werden mit einer mittleren Zuverlässigkeit von 0,87 simuliert. Dies stellt einen für heutige Kontextalgorithmen gute mittleren Zuverlässigkeitswert dar (vgl. [8]). Die Zuverlässigkeit der erzeugten Kontexte wird in der Simulation einer linearen zeitlichen Degression unterworfen, wie sie zum Beispiel von Schmidt in [144] zur Modellierung temporaler Einflüsse vorgeschlagen wurde. Die gewählte Degressionsrate liegt bei 0,1 pro Simulationsschritt. Des Weiteren wird angenommen, dass jeder Kontext durch

Segmentierung für maximal 5 Simulationsschritte im CSpace verfügbar ist. Simuliert wurden pro Simulationslauf 5000 Schritte. Artefakte vom Typ A sind im Durchschnitt in jedem vierten Simulationsschritt aktiv; das Artefakt vom Typ B dagegen ist in jedem Simulationsschritt aktiv und kann Kontexte verarbeiten. Eine Simulation besteht aus 100 aufeinander folgenden Simulationsläufen.

Ergebnisse

Durch den Aufbau dieses Szenarios werden im Durchschnitt pro Simulationsschritt 2,5 Kontexte des Typs a erzeugt. Damit das Artefakt B einen Kontext vom Typ b erzeugen kann, benötigt es allerdings mindestens 3 Kontexte a. Daher muss es auf ältere Kontexte zurückgreifen, die in früheren Simulationsschritten erzeugt wurden und die damit bereits einer temporalen Degression der Zuverlässigkeit in Bezug auf seine Anwendung unterliegen.

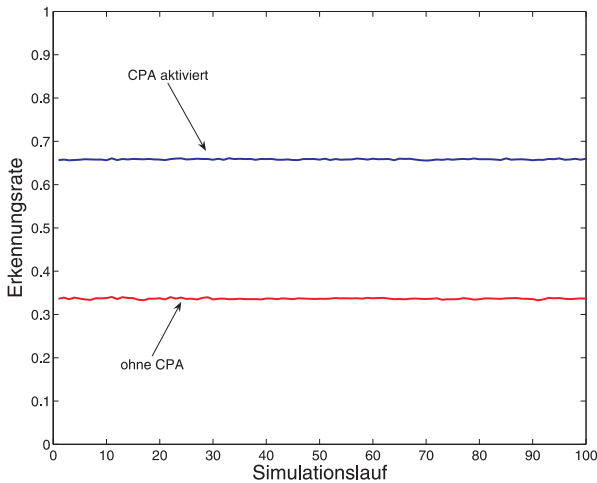


Abbildung 6.6. Erkennungsrate CPA: aktiviert (1 Schritt) und deaktiviert

Wird das CPA bei der Verarbeitung im Artefakt B nicht beachtet, werden zufällig Kontexte a ausgewählt, die im CSpace verfügbar sind. Da diese bis zu 5

Simulationsschritten alt sein können und damit eine entsprechend geringe Zuverlässigkeit haben, führt dies zu einer sehr geringen Erkennungsrate des Kontextes b im Artefakt B. Diese liegt über alle 100 Simulationsläufe im Mittel lediglich bei 0,3365.

Wird das CPA dagegen bei der Filterung von Kontexten beachtet, verbessert sich die Erkennungsrate, wie Abbildung 6.6 zeigt, auf einen durchschnittlichen Wert von 0,6583. Bei der Simulation mit aktiviertem CPA wurde der Qualitätsfilter so konfiguriert, dass das Artefakt B nur Kontexte a weiterverarbeitet, die maximal einen Simulationsschritt alt sind. Werden auch Kontexte a akzeptiert, die bereits 2 Simulationsschritte alt sind, liegt die durchschnittliche Erkennungsrate des Artefaktes B noch bei 0,5642.

Weitere Effekte der Qualitätsfilterung lassen sich erkennen, wenn man die Auswertung zweier Simulationsläufe mit und ohne CPA-Filterung vergleicht. Hier sind die Daten für das Artefakt B wiedergegeben. Die vollständigen Daten je eines Simulationslaufes sind in Anhang A zusammengestellt.

Simulation ohne CPA:

Stats for Artefact Typ_B_0:

Artefact type: 2

Total active steps:	5000
Total number of contexts consumed:	14994
Total number of contexts produced:	4998
Overall recognition rate:	0.3367674651174112
Mean relatedness of consumed Cs:	0.003897114401316082
Var of relatedness of consumed Cs:	4.254780673105989E-5

Simulation mit CPA:

Stats for Artefact Typ_B_0:

Artefact type: 2

Total active steps:	5000
Total number of contexts consumed:	7203
Total number of contexts produced:	2401
Overall recognition rate:	0.6562353088040962
Mean relatedness of consumed Cs:	0.0038178536720810848
Var of relatedness of consumed Cs:	4.112779654567796E-5

In beiden obigen Simulationsläufen wurden 5000 Verarbeitungsschritte simuliert. Das Artefakt B konnte in der Einstellung ohne CPA 14994 Kontexte kon-

sumieren und damit in fast jedem Simulationsschritt einen neuen Kontext produzieren. Insgesamt wurden so 4998 Kontexte vom Typ b produziert. Im Vergleich wurden bei der Simulation mit CPA nur 7203 Kontexte vom Typ a durch den Qualitätsfilter zur weiteren Verarbeitung zugelassen. Aus diesen konnten 2401 Kontexte des Typs b produziert werden. Das bedeutet, dass die Verwendung des Kommunikationskanals durch das Artefakt B, aufgrund der Filterung um etwa 48% verringert werden konnte, wobei die Kontexterkennungsrate des Artefaktes gleichzeitig von 0,3368 um über 51% auf 0,6562 verbessert hat.

Versuchsaufbau: CSA

Das CSA ist relevant für die Kontexterkennung, wenn Situationen anhand von längeren Kontextverarbeitungsketten erkannt werden. Aufeinander aufbauende Kontexte charakterisieren dabei die zu erkennende Situation.

Dieses Beispiel wird mit der Hilfe von 4 Artefakten simuliert, welche eine Verarbeitungskette bilden. Ein Artefakt vom Typ A produziert einen Kontext vom Typ a. Dieser wird von einem Artefakt vom Typ B konsumiert, das wiederum einen Kontext des Typs b produziert. Kontext b wird von Artefakt C konsumiert, das Kontext c produziert. Ein Artefakt vom Typ D, schließt die Verarbeitungskette ab. Es konsumiert Kontext c und produziert daraus Kontext d. Der Subskriptionsgraph entspricht in diesem Anwendungsbeispiel dem Ableitungsgraph. Beide stellen eine einfache Kette der Verarbeitung dar.

Die Zuverlässigkeit des Kontextalgorithmus von Artefakt A wird mit 0,87 simuliert. Bei der Ableitung aus nur einem Quellkontext wird diese Anfangszuverlässigkeit durch den simulierten Algorithmus im ableitenden Artefakt nicht weiter reduziert. Die temporale Degressionsrate liegt wieder bei 0,1 pro Simulationsschritt. Die Verweildauer eines Kontextes im CSpace wird mit maximal 7 Simulationsschritten angenommen. Simuliert wurden 5000 Schritte je Simulationslauf. Alle Artefakte wurden durchschnittlich in jedem vierten Schritt aktiviert. Eine Simulation besteht wieder aus 100 aufeinander folgenden Simulationsläufen.

Ergebnisse

In diesem Versuch werden die Kontexte a, b, c und d in einer Kette verarbeitet. Da nicht jeder dieser Kontexte in jedem Simulationsschritt erzeugt wird, kann die für

die Produktion des Kontextes d notwendige Verarbeitungskette, zwischen 4 und 28 Simulationsschritte in Anspruch nehmen. Dies führt zu einem signifikanten Unterschied in der temporalen Degression der Zuverlässigkeit der Kontexte in Abhängigkeit von der Dauer der Verarbeitungskette.

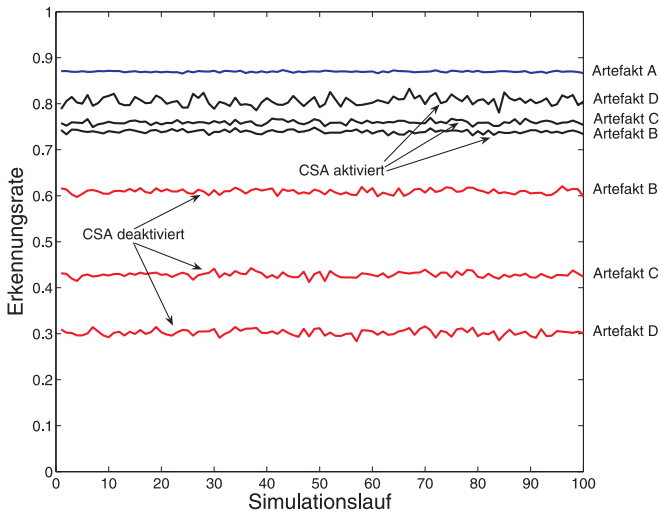


Abbildung 6.7. Erkennungsraten CSA: aktiviert (4 Schritte) und deaktiviert

Abbildung 6.7 zeigt den Vergleich der Erkennungsraten der simulierten Artefakte für eine Simulation mit deaktiviertem Filter auf das CSA und einer Simulation mit aktiviertem CSA-Filter. Es ist deutlich zu erkennen, wie die Erkennungsrate der Artefakte, ohne Filterung auf das CSA, entlang der Verarbeitungskette abnimmt. Die durchschnittliche Erkennungsrate von Artefakt B liegt ohne Filterung bei 0,6092, die von Artefakt C bei 0,4281 und die von Artefakt D lediglich noch bei 0,3025. Für die Vergleichsmessung mit aktivierter CSA-Filterung, wurde in dieser Simulation der Qualitätsfilter aller Artefakte so eingestellt, dass er Verarbeitungsketten bis zu einer Gesamtdauer von 4 Simulationsschritten akzeptiert. Damit verbessern sich die Kontexterkenneraten der Artefakte signifikant

auf durchschnittlich 0,7394 für Artefakt B, 0,7594 für Artefakt C und 0,8060 für Artefakt D.

Simulation ohne CSA:

General Context Stats

Total number of contexts consumed: 3009
Total number of contexts produced: 4271
Mean derive time: 7.854436689930223
Max internal context reliability: 0.999
Min internal context reliability: 0.06118118473419158
Mean internal context reliability: 0.5717944183419051

Simulation mit CSA:

General Context Stats

Total number of contexts consumed: 1184
Total number of contexts produced: 2449
Mean derive time: 2.700168918918917
Max internal context reliability: 0.999
Min internal context reliability: 0.5218457917658359
Mean internal context reliability: 0.810453042704471

Betrachtet man die obigen Ergebnisse von zwei Simulationsläufen mit dieser Konfiguration, erkennt man, dass die durchschnittliche Dauer aller Verarbeitungsketten in der Simulation mit der Aktivierung des Qualitätsfilters auf das CSA, deutlich, von 7,85 Simulationsschritten auf 2,70 Simulationsschritte, verringert wird. Auch hier kann, wie schon zuvor, beobachtet werden, dass die Anzahl der verarbeiteten und kommunizierten Kontexte sinkt, während deren durchschnittliche Zuverlässigkeit und damit auch die Erkennungsraten der Artefakte deutlich steigen (vgl. Anhang A)

In einem zweiten Versuch wurde zu Vergleichszwecken die Konfiguration der Qualitätsfilter der Artefakte noch einmal angepasst. Dabei wurden für die akzeptable Gesamtdauer der Kontextableitung für die Artefakte vom Typ B, C und D individuell zu 2, 3 und 4 Simulationsschritten gewählt.

Das Ergebniss dieser Simulation zeigt die Abbildung 6.8. Es ist zu erkennen, dass sich nun die Erkennungsraten der Artefakte B, C und D praktisch auf dem gleichen Niveau bewegen. Die durchschnittliche Kontexterkennungsrate des Artefaktes vom Typ B liegt nun bei 0,8262, die von Artefakt C bei 0,8138 und

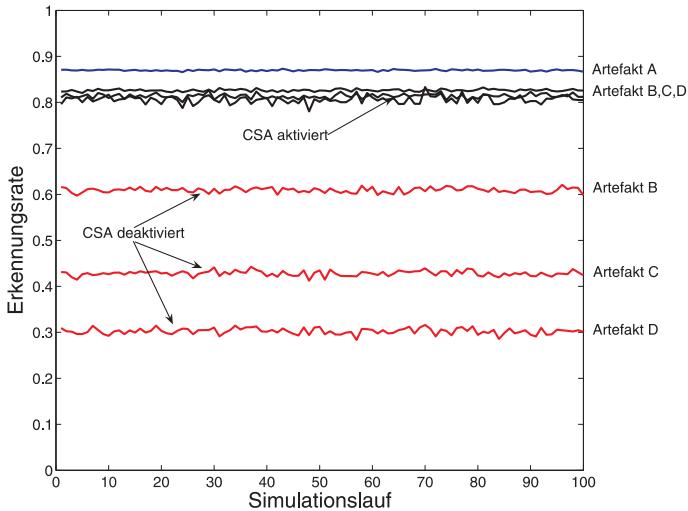


Abbildung 6.8. Erkennungsrates CSA: aktiviert (2, 3 und 4 Schritte)

schließlich die von Artefakt D bei 0,8060. Auch hier verbessert sich die Erkennungsrate der Artefakte B und C durch die optimalere Einstellung des Qualitätsfilters.

Zu beachten ist beim Vergleich der beiden Versuche, dass die Erkennungsrate des Artefaktes vom Typ D unverändert bleibt. Dies ist darauf zurückzuführen, dass in beiden Versuchen die Einstellung des Qualitätsfilters bei diesem Artefakt gleich waren. Da der Filter das Keimalter der Kontexte analysiert, ist eine Veränderung der Kontexterkennungsrate eines Artefaktes nur von seiner lokalen Einstellung bezüglich der Qualitätsfilterung auf das CSA abhängig. Die Einstellung von vorangestellten Artefakten bleibt ohne Auswirkungen.

6.2.2 Attribut LOC: Lokation

Das Kontextattribut LOC beschreibt die räumliche Herkunft eines Kontexts innerhalb einer ubiquitären Informationsumgebung. Fehler durch die Lokationszu-

ordnung entstehen, wie in dem motivierenden Beispiel aus Abschnitt 3.8.2, wenn mehrere gleiche Anwendungen, die lokale Kontexte verarbeiten, in einer ubiquitären Informationsumgebung an verschiedenen Orten gleichzeitig arbeiten. Das Anwendungsbeispiel war hier der AwarePen: Stifte an verschiedenen Orten, werden zum Schreiben an Wandtafeln verwendet. Die AwareCam-Anwendung, welche einer der Tafelinstallation durch ihre räumliche Anordnung zugeordnet ist, empfängt nun Stift-Kontexte von beiden Tafeln.

Versuchsaufbau

Dieses Anwendungsbeispiel wird nun in der context_sim Simulation umgesetzt. Um eine bessere Übersichtlichkeit zu erreichen, werden auch hier abstrakte Artefakt- und Kontexttypen verwendet. Die Abbildung 6.9 zeigt den Subskriptionsgraph, welcher der Simulation zu Grunde liegt.

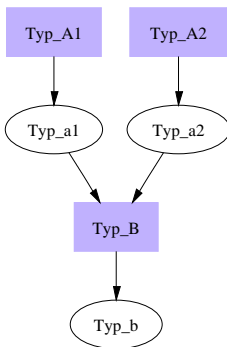


Abbildung 6.9. Simulation LOC: Subskriptionsgraph

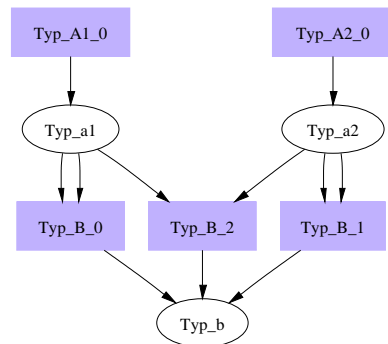


Abbildung 6.10. Simulation LOC: Ableitungsgraph

Die Simulation besteht aus drei Artefakten. Zwei Artefakte vom Typ A, welche sich an verschiedenen Orten (1 und 2) befinden produzieren Kontexte des Typs a. Diese Artefakte und ihre Kontexte sind zur einfacheren Unterscheidung mit dem Index ihres Ortes versehen. Das dritte Artefakt ist vom Typ B. Es befindet sich in Lokation 1 und produziert aus je zwei Kontexten vom Typ a einen neuen Kontext des Typs b.

Abbildung 6.10 zeigt den Ableitungsgraph dieser Simulation. Ihm ist zu entnehmen, dass das Artefakt vom Typ B drei verschiedene Möglichkeiten hat, den Kontext b zu produzieren. Das Artefakt B kann alternativ zwei Kontexte aus Lokation 1 oder Lokation 2 zum Kontext b weiterverarbeiten. Zusätzlich kann der Kontext b auch aus je einem Kontext aus Lokation 1 und einem aus Lokation 2 abgeleitet werden.

Für den Versuch wurden die Zuverlässigkeit der Kontextalgorithmen wie folgt festgelegt: Die Artefakte in Lokation 1 arbeiten mit einer mittleren Zuverlässigkeit von 0,87. Das Artefakt in Lokation 2 arbeitet bezüglich des abzuleitenden Kontextes b mit einer mittleren Zuverlässigkeit von 0,1. Dies bedeutet, sein Kontext beschreibt die Situation bezüglich Lokation 1 nur in 10% der Fälle korrekt. Das entspricht einer konstanten Degression der Kontextqualität bezüglich der Lokation (vgl. 3.8.2 und [144]). Die temporale Degression wurde deaktiviert, um Störeinflüsse zu vermeiden. Die Verweildauer eines Kontextes im CSpace wird mit einem Simulationsschritt angenommen. Simuliert wurden 5000 Schritte je Simulationslauf. Die Artefakte vom Typ A wurden durchschnittlich in 90% der Simulationsschritte aktiviert. Das Artefakt von Typ B war in jedem Schritt aktiviert. Eine Simulation besteht aus 100 aufeinander folgenden Simulationsläufen.

Ergebnisse

In diesem Versuch werden typgleiche Kontexte an verschiedenen Orten produziert. Für eine verarbeitendes Artefakt variiert die Zuverlässigkeit dieser Kontexte relativ zu seiner eigenen Lokation. Das bedeutet, Kontexte aus der gleichen Lokation haben für die simulierte Anwendung eine größere Zuverlässigkeit als Kontexte, die aus einer anderen Lokation stammen.

Wird das Attribut LOC, also die räumliche Herkunft der zu konsumierenden Kontexte bei der Verarbeitung nicht beachtet, verschlechtern die Kontexte der entfernten Lokation die Erkennungsrate des Artefaktes. Abbildung 6.11 zeigt die Ergebnisse einer Simulation mit aktiviertem Qualitätsfilter auf das Attribut LOC und einer Simulation mit deaktiviertem Filter auf dieses Attribut im Vergleich.

Wird die räumliche Herkunft der Kontexte bei der Verarbeitung nicht beachtet, liegt die durchschnittliche Kontexterkennungsrate des Artefaktes B bei 0,1865. Der Einfluss der Kontexte aus Lokation 2 ist deutlich, da diese die tatsächliche Situation in Lokation 1 nicht beschreiben können. Wird die Filterung von Kontexten aus anderen Lokationen aktiviert, verbessert sich die Erkennungsrate auf

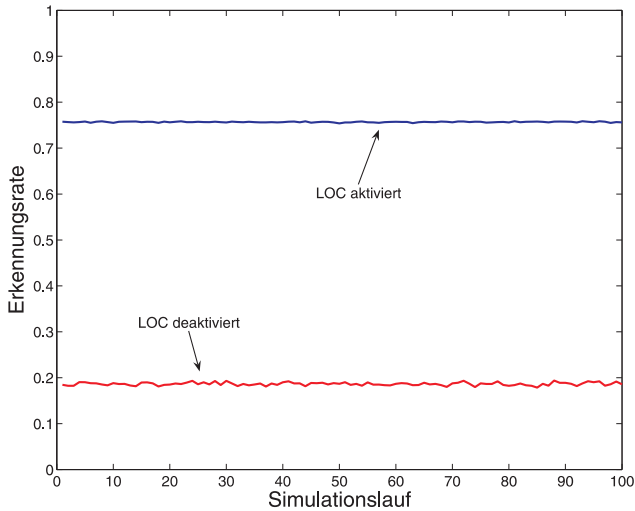


Abbildung 6.11. Erkennungsrate LOC: aktiviert und deaktiviert

einen durchschnittlichen Wert von 0,7568. Diese enorme Verbesserung der Erkennungsrate ist darauf zurückzuführen, dass auf der Basis der auf die lokale Situation in Lokation 2 bezogenen Kontexte vom Typ a, keine Aussage über die lokale Situation in Lokation 1 gemacht werden kann.

6.2.3 Attribut FCR: Verlässlichkeit

Das Attribut FCR beschreibt die Verlässlichkeit eines Kontextes aufgrund der Bewertung der Klassifikation des Kontextalgorithmus. Dabei wird die Klassifikationsleistung durch ein unabhängiges Fuzzy Inferenzsystem vorgenommen. FCR stellt das erste System zur quantitativen Bewertung der Verlässlichkeit von Kontexten in einer ubiquitären Informationsumgebung dar.

Versuchsaufbau

Die Verlässlichkeit von Kontexte hat in jeder Verarbeitungssituation in einer ubiquitären Informationsumgebung Einfluss auf die Erkennungsrate der Artefakte. Kontextalgorithmen klassifizieren ihre Eingaben zu Kontexten. Dabei ist für die Qualität der Klassifikation die Struktur der Eingabedaten ausschlaggebend.

Diese Simulation besteht aus 5 Artefakten. Sie bildet einen typischen Ausschnitt einer ubiquitären Informationsumgebung ab. Dabei wird ein Kontext (Typ e) in einem mehrstufigen Verarbeitungsprozess produziert. Abbildung 6.12 zeigt den Subskriptions- und Ableitungsgraph der Simulation.

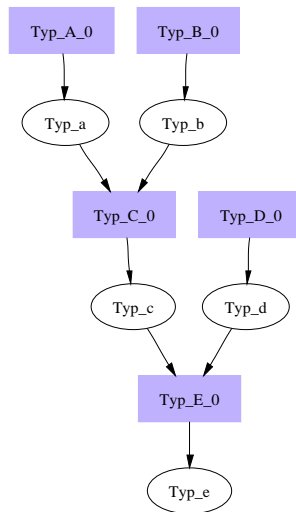


Abbildung 6.12. Simulation FCR: Subskriptions- und Ableitungsgraph

Die Artefakte A und B produzieren Kontexte erster Ordnung vom Typ a respektive Typ b. Diese werden von einem Artefakt vom Typ C konsumiert und zu einem Kontext vom Typ c aggregiert. Das Artefakt E konsumiert dann diesen Kontext höherer Ordnung zusammen mit einem weiteren Kontext erster Ordnung vom Typ d, der von einem Artefakt D produziert wird. Das Artefakt E aggregiert aus diesen beiden Eingabekontexten schließlich einen Kontext vom Typ e. Im Fokus der

Analyse stehen bei dieser Simulation die Erkennungsraten der kontextaggregierenden Artefakte C und E.

Die Zuverlässigkeit der Kontextalgorithmen aller Artefakte wird wieder mit einem mittleren Wert von 0,87 simuliert. Die Verweildauer eines Kontextes im CSpace wird auf 1 Simulationsschritt gesetzt. Simuliert wurden 5000 Schritte je Simulationslauf. Alle Artefakte werden in jedem Simulationsschritt aktiviert. Eine Simulation besteht aus 100 aufeinander folgenden Simulationsläufen.

Ergebnisse

Dieser Versuch bildet eine in ubiquitären Informationsumgebungen typische Baumstruktur der Kontextverarbeitung nach. Zu jedem der erzeugten Kontexte wird auf der Basis seiner, durch die Simulation zugeordneten Zuverlässigkeit, ein korrelierter FCR-Verlässlichkeitswert berechnet.

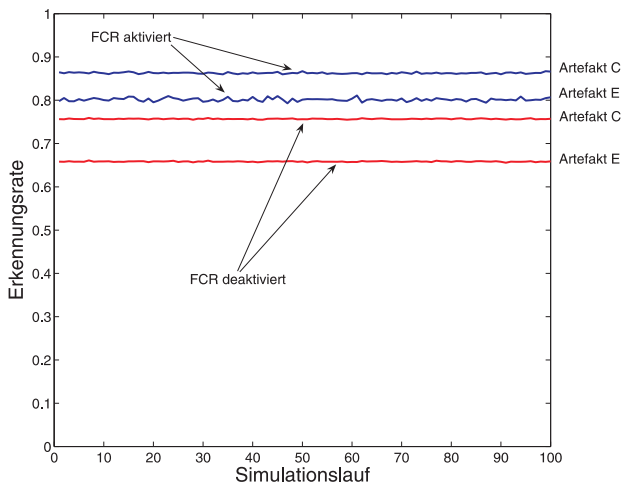


Abbildung 6.13. Erkennungsrate FCR: aktiviert und deaktiviert

Abbildung 6.13 zeigt die Ergebnisse einer Simulation mit aktiviertem FCR im Vergleich mit einer Simulation ohne Qualitätsfilterung. Ist kein Kontextmanage-

ment aktiviert, liegt die Kontexterkennungsrate des Artefaktes C bei 0,7568. Die Erkennungsrate von Artefakt D beträgt in diesem Fall im Mittel 0,6583. Für die Vergleichssimulation wurde der Qualitätsfilter der Artefakte C und D so konfiguriert, dass nur Kontexte akzeptiert werden, deren Verlässlichkeitswert mindestens 0,90 beträgt. Auf diese Weise werden Kontexte mit potentiell niedrigerer Zuverlässigkeit vor der Weiterverarbeitung ausgefiltert. Mit aktivierter FCR steigt die durchschnittliche Erkennungsrate des Artefaktes C um 10,6% auf einen Wert von 0,8628. Die Erkennungsrate von Artefakt D verbessert sich sogar um 14,32% auf einen Wert von 0,8015. Dies zeigt, dass Artefakte, welche Kontexte geringerer Ausgangszuverlässigkeit verarbeiten, von der Nutzung des Attribute FCR in höherem Maße profitieren können.

Simulation ohne FCR:

General Context Stats

Total number of contexts consumed: 19994
Total number of contexts produced: 24997
Mean derive time: 1.4999499849955051
Max internal context reliability: 0.9996547197270061
Min internal context reliability: 0.44634731740350775
Mean internal context reliability: 0.8052339051652415

Simulation mit FCR:

General Context Stats

Total number of contexts consumed: 948
Total number of contexts produced: 15474
Mean derive time: 1.2194092827004224
Max internal context reliability: 0.999948543733433
Min internal context reliability: 0.6780530165483398
Mean internal context reliability: 0.8691633926711935

Die obigen Ergebnisse von zwei einzelnen Simulationsläufen zeigen die Auswirkung der FCR-Filterung auf die Anzahl der verarbeiteten und kommunizierten Kontexte. Werden ohne Kontextmanagement noch 19994 Kontexte weiterverarbeitet, sinkt diese Zahl mit aktiviertem FCR-Filter auf 948 Kontexte. Gleichzeitig steigt die durchschnittliche Zuverlässigkeit aller, an der Simulation beteiligten Artefakte von etwa 0,8052 auf einen Wert von 0,8691, der nur minimal unter der durchschnittlichen Anfangszuverlässigkeit der Kontexte erster Ordnung in dieser Simulation liegt.

6.2.4 Attribut GRC: Verwandtschaft

Durch das Kontextattribut GRC können informationelle Abhängigkeiten zwischen Kontexten dargestellt und bewertet werden. Obwohl GRC nur auf lokalen Daten arbeitet, ermöglicht es den kontextverarbeitenden Artefakten einer ubiquitären Informationsumgebung Rückschlüsse auf die Qualität von Kontexten in Bezug auf diese Abhängigkeiten zu ziehen. So wird es mit der Nutzung dieses Attributes möglich, Einflüsse der globalen Verarbeitungsstruktur einer ubiquitären Informationsumgebung bei der Kontextverarbeitung zu berücksichtigen, ohne diese Struktur explizit zu kennen.

Die Leistungsfähigkeit des Attributes GRC wird hier anhand der bereits in Abschnitt 3.8.5 eingeführten Grundmuster der Kontextverarbeitung *Auffächerung und Multiplikation* und *zyklische Verarbeitung* evaluiert.

Versuchsaufbau: Auffächerung und Multiplikation

In diesem Versuch werden 5 Artefakte simuliert. Der Aufbau entspricht dem in Abbildung 3.12 eingeführten Schema. Die Abbildung 6.14 zeigt den dazugehörigen Simulationsgraph.

Ein Artefakt vom Typ A produziert hier einen Kontext a erster Ordnung, der parallel von 3 Artefakten vom Typ B zu einem Kontext des Typs b weiterverarbeitet wird. Ein Artefakt C fusioniert dann 3 Kontexte des Typs b zu einem Kontext vom Typ c.

Die Zuverlässigkeit der Kontextalgorithmen aller Artefakte wird auch bei diesem Versuch mit einem mittleren Wert von 0,87 simuliert. Für die Simulation der Auswirkung der Abhängigkeit von Kontexten auf die Erkennungsrate von Fusionsalgorithmen zur Kontextgewinnung, wird hier zwischen der tatsächlichen Abhängigkeit der Kontexte und der Zuverlässigkeit des Erkenners ein umgekehrt proportionaler Zusammenhang hergestellt. Die Verweildauer eines Kontextes im CSpace wird auf 3 Simulationsschritte gesetzt. Simuliert werden 5000 Schritte je Simulationslauf. Alle Artefakte werden in jedem Simulationsschritt aktiviert. Eine Simulation besteht wieder aus 100 aufeinander folgenden Simulationsläufen.

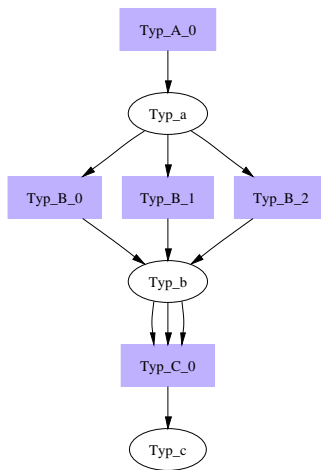


Abbildung 6.14. Simulation GRC: Auffächerung und Multiplikation

Ergebnisse

Durch den Aufbau dieses Versuchs werden Informationen, welche auf den Kontext a zurückgehen, durch die gleichartigen Artefakte vom Typ B vervielfacht und als Teil der Kontexte b in der Umgebung repliziert. Diese Kontexte b sind in hohem Maße voneinander abhängig, da sie alle auf denselben Ausgangskontext zurückgehen.

Werden durch das Artefakt C nun Kontexte vom Typ b fusioniert, welche alle auf demselben Kontext a beruhen, sinkt die Erkennungsrate von C, wogegen sie ansteigt, wenn die fusionierten Kontexte unabhängig sind.

Die Abbildung 6.15 zeigt die Ergebnisse zweier Simulationen im Vergleich. Bei der ersten Simulation war der Qualitätsfilter auf das Attribut GRC deaktiviert. In diesem Fall liegt die durchschnittliche Kontexterkennungsrate des Artefaktes C bei 0,6061.

Bei der zweiten, in Abbildung 6.15 gezeigten, Simulation wurde der Qualitätsfilter für das Attribut GRC mit einem maximalen Verwandtschaftsgrad von 0,5 konfiguriert. Dadurch werden Kontexte, welche auf denselben Ausgangsinforma-

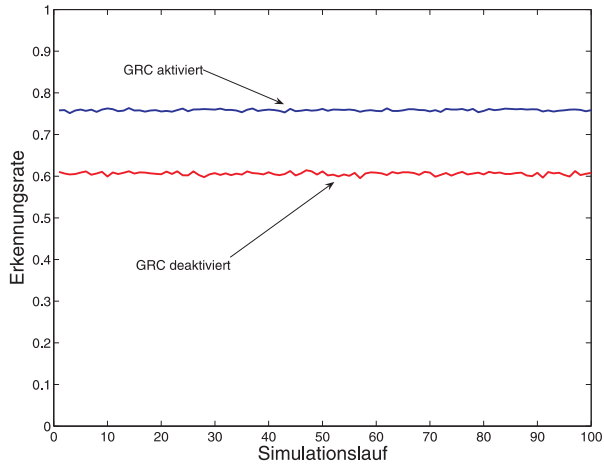


Abbildung 6.15. Erkennungsrate GRC: aktiviert und deaktiviert

tionen beruhen, ausgefiltert. Die durchschnittliche Erkennungsrate des Artefaktes C steigt dadurch auf einen Wert von 0,7585.

Simulation ohne GRC:

Stats for Artefact Typ_C_0:

Artefact type: 3

Total active steps:	5000
Total number of contexts consumed:	14994
Total number of contexts produced:	4998
Overall recognition rate:	0.6075336458767863
Mean relatedness of consumed Cs:	0.3868458494508915
Var of relatedness of consumed Cs:	0.08781520814236893

Simulation mit GRC:

Stats for Artefact Typ_C_0:

Artefact type: 3

Total active steps:	5000
---------------------	------

```
Total number of contexts consumed: 7386
Total number of contexts produced: 2462
Overall recognition rate: 0.7584927144079863
Mean relatedness of consumed Cs: 0.00375485152089539
Var of relatedness of consumed Cs: 4.0876524052520745E-5
```

Analysiert man die obigen Ergebnisse zweier einzelner Simulationsläufe, sieht man, dass bei dem Versuch ohne aktiviertes GRC, der mittlere Verwandtschaftsgrad der verarbeiteten Kontexte bei 0,3868 liegt. Das heißt über ein Drittel aller verarbeiteten Kontexte, gehen auf dieselben Ausgangsdaten zurück. Bei dem Versuch mit Qualitätsfilterung fällt der mittlere Verwandtschaftsgrad auf einen Wert von 0,0037. Dies bedeutet, dass die nun verarbeiteten Kontexte unabhängig voneinander sind.

In diesem Beispiel wurde der Qualitätsfilter auf den Grenzwert von 0,5 für den maximal akzeptablen Verwandtschaftsgrad von Kontexten konfiguriert. Da hier Kontexte aber entweder alle *direkt* auf dieselbe Quelle zurückgehen (der Verwandtschaftsgrad dieser Kontexte ist nahezu 1,0) oder vollständig unabhängig sind, hätte eine weniger restriktive Filterung, zum Beispiel mit einem Wert von 0,9, vergleichbare Ergebnisse geliefert.

Versuchsaufbau: zyklische Verarbeitung

Im Versuch zur zyklischen Verarbeitung, werden 4 Artefakte simuliert. Hier entspricht der Aufbau des Versuchs der schematischen Darstellung in Abbildung 3.13. Der korrespondierende Simulationsgraph ist in Abbildung 6.16 zu sehen.

Das Artefakt vom Typ D produziert einen Kontext des Typs c. Dieses Artefakt wird im Durchschnitt nur jeden einhundertsten Simulationsschritt aktiviert. Der von D produzierte Kontext c initiiert eine zyklische Verarbeitung: Der Kontext c wird von Artefakt A konsumiert, welches daraus einen Kontext a ableitet. Dieser wird von einem Artefakt vom Typ B aufgenommen und zu einem Kontext vom Typ b weiterverarbeitet, bevor dieser schließlich von Artefakt C konsumiert wird, welches wiederum einen Kontext des Typs c produziert und so den Kreis schließt. Die Artefakte A, B und C in diesem Versuch werden in jedem Simulationsschritt aktiviert.

Die Zuverlässigkeit des Kontextalgorithmus des Artefaktes D wird bei diesem Versuch wieder mit einem mittleren Wert von 0,87 simuliert. Für die Simulation der Auswirkung der Abhängigkeit von Kontexten auf die Erkennungsrate der

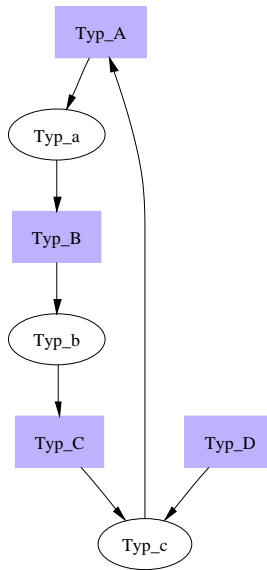


Abbildung 6.16. Simulation GRC: zyklische Vererbung

Artefakte bei der Weiterverarbeitung einzelner Kontexte, wird hier von einer linearen Degression der Zuverlässigkeit ausgegangen. Die Degressionsrate beträgt in diesem Versuch 0,1% der Ausgangszuverlässigkeit. Sie ist damit extrem gering gewählt.

Die Verweildauer eines Kontextes im CSpace wird mit einem Simulationsschritt angenommen. Der Historienspeicher der Artefakte hat 10 Einträge. Simuliert werden wieder 5000 Schritte je Simulationslauf. Eine Simulation besteht auch hier aus 100 aufeinander folgenden Simulationsläufen.

Ergebnisse

Da das Artefakt D, welches als einziges in diesem Versuch dem System unabhängige Informationen bereitstellt, nur selten aktiv ist, aber die weiterverarbeitenden Artefakte, welche den eigentlichen Kreis bilden, dagegen in jedem Simulations-

schritt aktiviert werden, etabliert sich ohne Qualitätsfilterung eine zyklische Verarbeitung, der nur sehr selten neue Informationen zugeführt werden. Die Kontexte, die sich in diesem Fall in dem Zyklus befinden sind nach ihrem ersten Umlauf in hohem Maße voneinander abhängig. Damit sinkt die Erkennungsrate der an diesem Kreis beteiligten Artefakte von dem anfänglichen Wert von 0,87 kontinuierlich, bis neue Informationen in den Kreis einfließen.

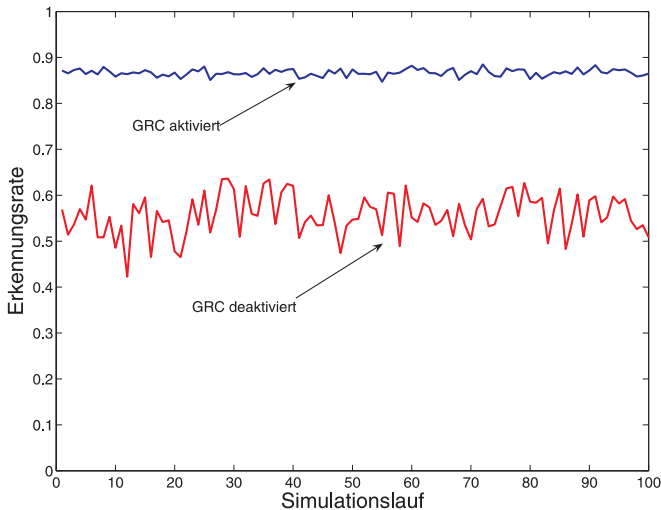


Abbildung 6.17. Erkennungsrate GRC: aktiviert und deaktiviert

Betrachtet man die Kontexterkennungsrate eines der Artefakte, das Teil des Kreises ist, Abbildung 6.17 zeigt zum Beispiel die Erkennungsrate von Artefakt C, erkennt man, dass die Erkennungsrate ohne Qualitätsfilterung stark schwankt. Sie liegt im Mittel über alle Simulationsläufe bei einem Wert von 0,5576.

Die starken Schwankungen beruhen auf den unterschiedlichen absoluten Anfangszuverlässigkeiten des von Artefakt D produzierten Kontextes c. Diese schwanken wie oben beschrieben, um einen Mittelwert von 0,87. Dabei wird der konkrete Zuverlässigkeitswert eines einzelnen Kontextes, wie bei allen hier vorgestellten Simulationen, mit Hilfe einer Gaußfunktion erzeugt. Diese hat ihren Mittelwert

bei dem eingestellten Vorgabewert – hier 0,87 – und eine Standardabweichung von 0,05. Die Schwankung in der Anfangszuverlässigkeit wird durch den hier gezeigten Versuchsaufbau stark verstärkt.

Die zweite Kurve in Abbildung 6.17 zeigt die Erkennungsrate von Artefakt C mit aktiviertem Qualitätsfilter. In diesem Versuch wurden die Artefakte, welche den Verarbeitungszyklus bilden, so konfiguriert, dass ihr Qualitätsfilter eine Verarbeitung von Kontexten mit einem Verwandtschaftsgrad von mehr als 0,5 nicht gestattet. Der Verwandtschaftsgrad wird in diesem Fall, da nur ein einzelner Eingangskontext verarbeitet wird, zwischen dem Genom dieses Kontextes und den Kontextgenomen im Historienspeicher des Artefaktes berechnet.

In dieser Konfiguration steigt die Kontexterkennungsrate des Artefaktes vom Typ C auf einen Wert von 0,8667.

Simulation ohne GRC:

Stats for Artefact Typ_C_0:

Artefact type: 3

Total active steps:	5000
Total number of contexts consumed:	4767
Total number of contexts produced:	4767
Overall recognition rate:	0.5081804581474955
Mean relatedness of consumed Cs:	0.8095797496678541
Var of relatedness of consumed Cs:	0.15393581206897283

General Context Stats

Total number of contexts consumed:	14304
Total number of contexts produced:	14345
Mean derive time:	683.3587108501164
Max internal context reliability:	0.999
Min internal context reliability:	0.1288437726973299
Mean internal context reliability:	0.5097629374020071

Simulation mit GRC:

Stats for Artefact Typ_C_0:

Artefact type: 3

Total active steps:	5000
Total number of contexts consumed:	56
Total number of contexts produced:	56
Overall recognition rate:	0.8649593397028101
Mean relatedness of consumed Cs:	0.018452380952380956
Var of relatedness of consumed Cs:	0.0178553391053391

General Context Stats

```
Total number of contexts consumed: 168
Total number of contexts produced: 224
Mean derive time: 1.9999999999999993
Max internal context reliability: 0.9756014921103677
Min internal context reliability: 0.7497204081277599
Mean internal context reliability: 0.866258944358858
```

Der obige Vergleich von zwei Simulationsläufen zeigt bei diesem Versuch deutlich, wie sich die Filterung auf die Kontextverarbeitung auswirkt. Ist GRC nicht aktiviert, verbleiben Kontexte lange in dem etablierten Zyklus. Dies zeigt sich vor allem an der durchschnittlichen Ableitungsdauer von über 680 Simulationsschritten für einen Kontext. Der Verwandtschaftsgrad der von Artefakt C verarbeiteten Kontexte liegt dabei etwa bei 0,8096. Außerdem wurden während dieses Simulationslaufes je über 14000 Kontexte konsumiert und produziert.

Wird der Qualitätsfilter für das Attribut GRC dagegen aktiviert, werden Kontexte automatisch verworfen, bevor sich der Verarbeitungszyklus ausbilden kann. Dies führt dazu, dass jedes Artefakt nur die Anzahl an Kontexten konsumiert und auch produziert, die das Artefakt D als neue Information im Laufe der Simulation bereitstellt. Neben dem deutlichen Anstieg der Kontexterkennungsrate, ist eine Reduktion des Verwandtschaftsgrades auf einen Wert von etwa 0,0185 zu beobachten. Dies belegt, dass die verarbeiteten Kontexte jetzt unabhängig sind. Durch die Unterbrechung des Kreises reduziert sich außerdem die durchschnittliche Ableitungszeit für einen Kontext auf ungefähr 2 Simulationsschritte.

6.2.5 Gesamtszenario

Das nun untersuchte Gesamtszenario führt die zuvor analysierten Versuche zusammen. Auf diese Weise entsteht eine realitätsnahe Simulation einer komplexen ubiquitären Informationsumgebung mit einer Vielzahl von interagierenden Artefakten.

Im Gegensatz zu den Einzelversuchen ist es hier allerdings kaum möglich, die Einflüsse der einzelnen Attribute getrennt zu quantifizieren. Mehrere Faktoren verhindern dies: Die Fehlersituationen, welche durch den Einsatz der einzelnen Attribute im Rahmen des Kontextmanagements adressiert werden, sind in einer

realitätsnahen ubiquitären Informationsumgebung nicht gleich verteilt. Damit ist auch der Einfluss auf die Erkennungsrate dieser Fehlerquellen unterschiedlich. Hinzukommt, dass bestimmte Fehlersituationen nur durch die Kombination mehrerer Attribute optimal gelöst werden können. In diesen Fällen beeinflusst, bei einer separaten Betrachtung der Attributauswirkungen, die Reihenfolge der Aktivierung einzelner Attribute das Ergebnis. Ein weiterer Faktor, welcher die quantitative Zuordnung erschwert, sind Synergieeffekte. Diese äußern sich darin, dass Situationen auftreten, in denen das Aktivieren einzelner Attribute in Summe zu deutlich geringeren Verbesserungen der Erkennungsrate führen kann, als die Aktivierung der entsprechenden Kombination von Kontextattributen. Außerdem muss bei der Betrachtung einzelner Artefakte in der Umgebung beachtet werden, dass diese in unterschiedlichem Maße vom Einsatz der einzelnen Attribute profitieren. Zum Beispiel können Artefakte, welche ausschließlich Kontexte erster Ordnung produzieren, generell nicht von dem hier vorgestellten Qualitätssicherungssystem für Kontexte profitieren, da dieses eigens für die Verbesserung der Erkennungsrate von Kontexten höherer Ordnung entwickelt wurde. Dies schlägt sich vor allem bei der Betrachtung der durchschnittlichen Erkennungsrate der gesamten ubiquitären Informationsumgebung nieder, da hier Artefakte, welche Kontexte erster Ordnung produzieren, immer mit der gleichen durchschnittlichen Erkennungsrate, in das Ergebnis eingehen. Diese durchschnittliche Erkennungsrate entspricht dabei der für die Simulation vorgegebenen Zuverlässigkeit des Kontextalgorithmus dieser Artefakte.

Versuchsaufbau

Dieser Versuch besteht aus 98 Artefakten in zwei Räumen. Er bildet zwei unabhängige AwareOffice Umgebungen in einem Gebäude nach. Simuliert werden dabei je Besprechungsraum: 12 Stühle, 6 Fenster, 1 Tür, 10 Tassen, 1 Projektor, 1 Wandtafel, 4 Stifte, 1 Schwamm, 9 PDAs, 1 Digitalkamera, 1 interaktives Türschild, 1 Klimaanlage und 1 Tisch. Dies entspricht, bis auf die Klimaanlage, den Artefakten, welche auch in der realen AwareOffice Testumgebung eingesetzt werden können.

Der Versuch bildet eine Umgebung bestehend aus diesen Besprechungsräume nach, welche sich in einem Gebäude befinden und über die dort vorhandene Infrastruktur untereinander verbunden sind. Der Subskriptionsgraph dieser Simulation ist dem Anhang A.5.3 zu entnehmen.

Stifte, Schwamm, Tafel und Kamera bilden den Anwendungskomplex Aware-Cam. Wird mit den Stiften an der Tafel geschrieben, kann die Kamera ein Bild von der Tafel aufnehmen und es an die Teilnehmer der Besprechung versenden. Wird der Schwamm benutzt, bevor eine Aufnahme des Tafelbildes gelungen ist, kann der Benutzer gewarnt werden.

Stühle, Tische, Tassen und Projektoren können die Anwesenheit von Benutzern im AwareOffice detektieren. Durch die Kombination ihrer Kontexte ist es möglich, die Aktivität im AwareOffice abzuschätzen.

Türschilder und PDAs sind in dieser Simulation in der Lage, diese Aggregation auf verschiedene Weisen durchzuführen. Sie stellen einen Besprechungskontext bereit, können diesen aber auch selbst verarbeiten, um zum Beispiel im Falle der PDAs, ihre Alarmierungsfunktion an die Situation anzupassen.

Die Klimaanlage schließlich aggregiert den Besprechungskontext mit dem Zustand der Fenster und steuert abhängig von dieser Information ihre Leistung.

Die Zuverlässigkeit aller Kontextalgorithmen wird auch bei diesem abschließenden Versuch mit einem mittleren Wert von 0,87 simuliert. Für die Simulation der Auswirkung der Abhängigkeit von Kontexten auf die Erkennungsrate wird hier wieder von einer linearen Degression der Zuverlässigkeit ausgegangen. Die Degressionsrate beträgt 0,1% der Ausgangszuverlässigkeit. Für die Simulation der Auswirkung der Abhängigkeit von Kontexten auf die Erkennungsrate von Fusionsalgorithmen, wird zwischen der tatsächlichen Abhängigkeit der Kontexte und der Zuverlässigkeit des Erkenners ein umgekehrt proportionaler Zusammenhang hergestellt. Die Zuverlässigkeit der Kontexte wird in der Simulation einer linearen zeitlichen Degression unterworfen. Die gewählte Degressionsrate liegt hier bei 0,1 pro Simulationsschritt. Die Zuverlässigkeit von Kontexten aus einer, aus Sicht der Anwendung falschen Lokation, wird mit einem Faktor von 0,1 bewertet.

Die maximale Verweildauer eines Kontextes im CSpace wird mit 8 Simulationsschritten angenommen. Der Historienspeicher der Artefakte hat 10 Einträge. Simuliert werden auch hier 5000 Schritte je Simulationslauf. Und eine Simulation besteht aus 100 aufeinander folgenden Simulationsläufen.

Ergebnisse

Dieser Versuch simuliert eine relativ kleine ubiquitäre Informationsumgebung. Trotzdem ist die Komplexität der Zusammenhänge in der Verarbeitung von Kontextdaten bereits in diesem einfachen Beispiel so hoch, dass einzelne Einflüsse nicht mehr dediziert beobachtet werden können.

Um trotzdem einen Eindruck von der Auswirkung des Qualitätsmanagementsystems auf die Komponenten der simulierten Umgebung geben zu können, zeigt die Abbildung 6.18 einen Vergleich der Erkennungsraten aller Artefakte mit und ohne aktives Kontextmanagement.

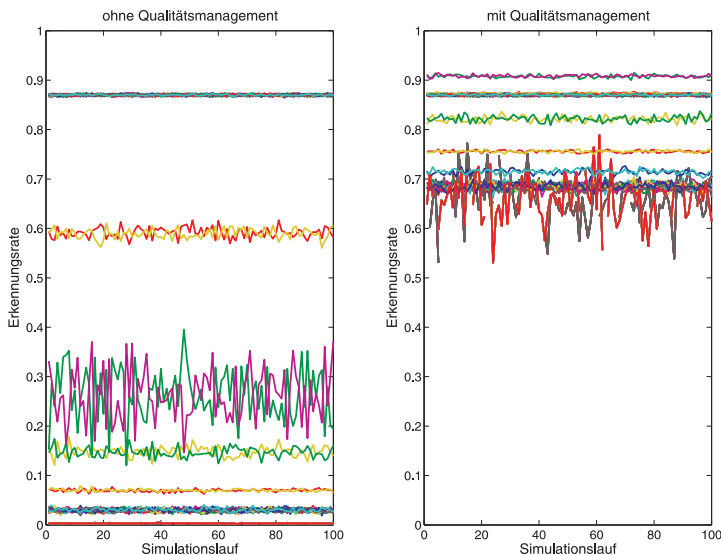


Abbildung 6.18. Erkennungsraten: links ohne, rechts mit Kontextmanagement

In der Abbildung erkennt man bei einer Erkennungsrate von etwa 0,87 die Artefakte, welche Kontexte erster Ordnung produzieren. Ihre Lage verändert sich durch das Aktivieren des Qualitätsmanagementsystems nicht. Des Weiteren bringt diese Abbildung zwei wichtige Erkenntnisse: Erstens profitieren alle

Artefakte, welche Kontexte höherer Ordnung produzieren von dem Qualitätsmanagement. Zweitens kann in der rechten Hälfte der Abbildung beobachtet werden, dass nun offenbar kontextfusionierende Artefakte in der Lage sind, Kontexte mit einer höheren Erkennungsrate als der ursprünglichen Grundzuverlässigkeit der Ausgangskontexte, zu produzieren. Das bedeutet, dass mit aktivem Qualitätsmanagement eine reale Verbesserung der Zuverlässigkeit durch Kontextfusion möglich wird.

Für diesen Versuch wurden die Qualitätsfilter der Artefakte wie folgt konfiguriert: Für alle Artefakte CPA: maximal 2 Simulationsschritte, CSA: maximal 4 Simulationsschritte, LOC: nur Kontexte aus dem gleichen Raum und FCR: Verlässlichkeitswert von mindestens 0,90. Zusätzlich wurde GRC für Artefakte, welche Kontexte aggregieren auf einen maximal Wert von 0,5 gesetzt. Für Artefakte, welche Kontexte fusionieren, wurde GRC auf einen maximalen Verwandtschaftsgrad von 0,1 konfiguriert.

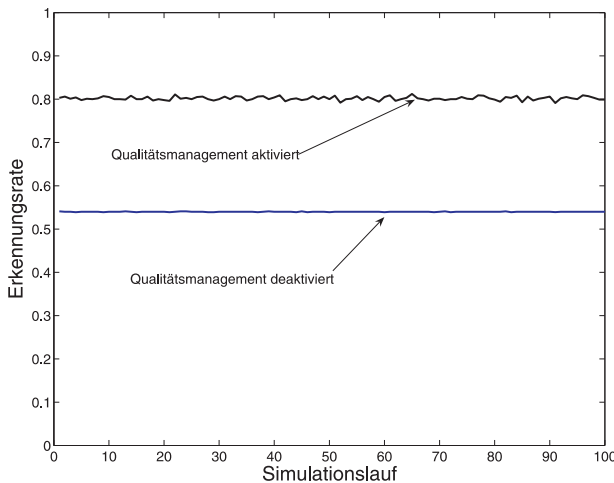


Abbildung 6.19. Mittlere Erkennungsraten: ohne und mit Kontextmanagement

In Abbildung 6.19 sind die durchschnittlichen Erkennungsraten der Gesamtumgebung gegeneinander aufgetragen. Im Mittel beträgt diese bei aktiviertem

Kontextmanagement 0,8019. Dagegen liegt sie ohne das Kontextmanagementsystem lediglich bei einem mittleren Wert von 0,5399.

Hier ist zu beachten, dass die Artefakte, welche ausschließlich Kontexte erster Ordnung produzieren, in das Ergebnis eingehen, obwohl diese von der Verbesserung der Erkennungsrate durch das Kontextmanagementsystem nicht betroffen sind. Trotzdem liegt die Gesamterkennungsrate im Durchschnitt mit aktivierter Qualitätskontrolle um 26,2% höher. Außerdem muss beachtet werden, dass dieses Ergebnis erreicht wird, ohne dass die Leistung der Kontextalgorithmen in der Umgebung verbessert wurde.

6.3 Feldversuch: AwareOffice

In diesem Abschnitt werden die Ergebnisse der Evaluierung des Qualitätsmanagementsystems vorgestellt, welche auf Versuchen mit realen Anwendungen und Artefakten in der Testumgebung des AwareOffice beruhen.

Bei der Evaluierung in einer realen ubiquitären Informationsumgebung besteht die Hauptschwierigkeit in der Messung und Zuordnung der Erkennungsraten von Artefakten und Anwendungen. Im Gegensatz zur simulativen Umgebung stehen in der realen Testumgebung keine technischen Hilfsmittel bereit, die eine automatische Registrierung der korrekten Erkennung eines Kontextes erlauben. Die Korrektheit der Kontexterkennung muss hier in jedem Fall von einem menschlichen Beobachter festgestellt und für die Auswertung dokumentiert werden. Die Feststellung der Korrektheit beruht dabei auf der Einschätzung des Beobachters, der seinen subjektiven Eindruck der Situation in Beziehung zu der, durch die kontextsensitive Anwendung erkannten Situation setzen muss. Daher ist die Analyse komplexer Szenarien mit vielen gleichzeitig agierenden Anwendungen in der realen Umgebung extrem schwierig und potentiell mit großen Fehlern behaftet.

Aus diesem Grund wurden für die quantitative Evaluierung des Kontextmanagementsystems einzelne Anwendungen aus dem AwareOffice ausgewählt und die Auswirkungen der Qualitätsfilterung an diesen Anwendungen im Rahmen einfacher Szenarien analysiert. Dabei wurde, wie schon in der simulativen Evaluierung, die Wirkung der verschiedenen Kontextattribute getrennt voneinander untersucht. Da in der realen Testumgebung keine durchschnittliche Erkennungsrate über alle aktiven Anwendungen parallel berechnet werden kann und einzelne Anwendungen aus dem AwareOffice nicht geeignet sind Szenarien zu erstellen,

welche alle von den Kontextattributen adressierten Fehlerfälle abdecken, mussten die Attribute einzeln evaluiert werden.

6.3.1 Feldversuch: CA und LOC

Für die Evaluierung der Auswirkung der Attribute Kontextalter und räumliche Herkunft von Kontexten wurden die bereits für die Simulation in context_sim eingesetzten Szenarien im AwareOffice nachgebildet.

Das CPA wurde anhand des in Abschnitt 6.2.1 vorgestellten Reinigungsszenarios evaluiert. Hierzu wurde die Anwendung des interaktiven Türschildes so angepasst, dass sie einen Besprechungskontext unter ausschließlicher Benutzung der Aktivität der Stühle im AwareOffice ableitet. Für den Versuch wurde dann eine Besprechung im AwareOffice abgehalten. Später wurde der Raum gesaugt.

Das interaktive Türschild wurde so programmiert, dass eine Besprechung erkannt wird, wenn mindestens drei der Stühle im AwareOffice aktiv sind. Die Aktivierung der Stühle erfolgt auf der Basis der Daten eines Bewegungssensors. Dieser kann nicht unterscheiden, ob eine Person auf dem Stuhl sitzt und diesen so in leichte Bewegung versetzt, oder ob der Stuhl auf eine andere Weise bewegt wird.

Das Ergebnis dieses Versuches ist es, dass das Türschild ohne aktivierte Filterung des CPA nicht in der Lage ist, die Besprechungssituation korrekt vom Saugen des Raumes zu unterscheiden. Die Erkennungsrate für die Besprechung in ersten Versuchsteil lag durchschnittlich bei 0,8. Im zweiten Teil des Versuches wurde dagegen fälschlich ebenfalls eine Besprechung erkannt. Die Erkennungsrate des Besprechungskontextes über den gesamten Versuch ergab sich damit im Durchschnitt zu einem Wert von 0,45.

Bei aktiver Filterung wurde der Kontext dagegen auch im zweiten Versuchsteil überwiegend korrekt erkannt, was zu einer Gesamterkennungsrate von durchschnittlich 0,78 für diesen Versuch führte.

Für die Evaluierung des CSA wurde die hier vorgestellte Anwendung erweitert: Der Besprechungskontext wird in diesem Versuch mit den Kontexten der AwarePens zu einem Vortragskontext aggregiert. Ein Vortrag ist dabei definiert als eine Situation, in der von dem interaktiven Türschild eine Besprechung erkannt wird und mindestens einer der AwarePens aktiv ist. Der AwarePen ist immer aktiv, wenn mit ihm geschrieben oder gestikuliert wird. Diese beiden Kontexte werden dann von einem PDA aggregiert und so ein Vortragskontext erzeugt.

Für diesen Versuch wurde im AwareOffice zuerst ein Vortrag gehalten. Das heißt mehrere Zuhörer waren anwesend, während die vortragende Person einen AwarePen benutzte. Anschließend wurde eine Besprechung ohne Benutzung des AwarePens abgehalten. Und im direkten Anschluss daran, wurde dann von einer einzelnen Person an die Tafel geschrieben, ohne dass sich weitere Teilnehmer im AwareOffice befunden haben.

Bei aktivierter Qualitätsfilterung wurde eine Erkennungsrate des Vortrags- und Besprechungskontextes von 0,72 erreicht. Ohne die Beachtung des CSA wurde die Besprechung mit darauf folgendem Tafelanschrieb in etwa der Hälfte der Fälle ebenfalls als Vortrag interpretiert.

Die Wirkung des Attributes LOC wurde anhand der AwareCam Anwendung evaluiert. Zu diesem Zweck wurde gleichzeitig an den zwei Weißwandtafeln im AwareOffice und im Aufenthaltsbereich der Institutes Anschnitte vorgenommen. Die Erkennungsrate wurde dann für die AwareCam Anwendung im AwareOffice analysiert. Wie zu erwarten war, hängt diese stark von dem Verhältnis ab, in dem an den beiden Orten geschrieben wird. Das Ergebnis zeigt, dass ohne Filterung der räumlichen Herkunft, alle Kontexte aus dem Aufenthaltsbereich durch die AwareCam falsch interpretiert wurden. Dagegen wurde mit aktiviertem Filter kein Kontext aus dem Aufenthaltsbereich von der AwareCam akzeptiert. Die Erkennungsrate der AwareCam lag in diesem Fall durchschnittlich bei 0,78.

Beide hier untersuchten Attribute, CA und LOC, adressieren Fehlersituationen, welche direkt zu einer vollständigen Fehlinterpretation der Kontexte durch die konsumierende Anwendung führen. Eine quantitative Beurteilung der Verbesserung der Erkennungsrate bedarf daher weiterer Interpretation: Ist die Filterung auf diese Attribute nicht aktiviert, können die entsprechenden Fehlersituationen nicht erkannt und behandelt werden. Abhängig von der untersuchten Situation kann daher die Verbesserung der Erkennungsrate einer Anwendung stark schwanken.

Daher wurden die Versuche hier so ausgelegt, dass sie sowohl die Fehlersituation, als auch dasselbe Szenario ohne die Fehlersituation abdecken. Beide Teilversuche flossen dann – möglichst gleich gewichtet – in die Berechnung ein. Die erzielten Ergebnisse decken sich im Wesentlichen mit denen aus der Simulation. Zu beachten ist hier bei einem direkten Vergleich der Werte, dass sich die Basiserkennungsrate der realen Artefakte, von dem, bei der Simulation zu Grunde gelegten Referenzwert aus der Literatur von 0,87, unterscheiden. Dies führt in der Folge in den Erkennungsrate von Artefakten, welche diese Kontexte konsumieren und weiterverarbeiten, zu abweichenden Absolutwerten.

6.3.2 Feldversuch: FCR

Die Evaluierung der Auswirkung des Attributes FCR ist weniger problematisch. Das Attribut dient der Sicherung der Verlässlichkeit von Kontexten. Da jeder Kontextalgorithmus als Klassifikator, mit einer gewissen Unzuverlässigkeit behaftet ist, die im Allgemeinen von der Charakteristik der zu klassifizierenden Daten abhängt, kann hier die Erkennungsrate einer einzelnen Anwendung betrachtet werden. Außerdem kann stets die gleiche Situation untersucht werden. Eine Trennung in zwei Versuchsteile wie bei den Attributen CA und LOC ist nicht nötig, da die Fehlersituation hier grundsätzlich immer besteht und auch nicht vermieden werden kann.

Für die Evaluierung wurde hier wieder die AwareCam Anwendung und ihre Erkennungsrate der Stiftkontexte betrachtet. Die Teilnehmer wurden gebeten, mit einem AwarePen das Panagramm „*The quick brown fox jumps over the lazy dog. 1234567890*“ an eine Weißwandtafel zu schreiben. Der Versuchsablauf bestand für jeden Probanden aus je sechs Schriftproben, die in einem kontinuierlich beobachteten Versuch genommen wurden. Zu Beginn lag der Stift dabei ruhig auf einem bereitgestellten Tisch. Die Versuchsperson sollte dann den Stift aufnehmen und das Panagramm schreiben und den Stift sodann wieder auf dem Tisch ablegen. Dieser Vorgang wurde sechs Mal ohne Unterbrechung der Aufzeichnung wiederholt. Während des Versuches wurden die Kontexte von einem Versuchsleiter annotiert, so dass bei der späteren Auswertung des Versuches die Erkennungsrate des Stiftkontextes errechnet werden konnte.

Der Versuch umfasste die Schriftproben von sieben unterschiedlichen Personen. Als Stift kam ein AwarePen Artefakt zum Einsatz, welches als Kontextalgorithmus ein neuronales Netzwerk¹ einsetzt. Der Kontextalgorithmus wurde vor dem Versuch mit Hilfe von unabhängigen Trainings- und Testdaten einer der Testpersonen trainiert.

Als Schwellwert für die Filterung des Verlässlichkeitsmaßes wurde ein Wert von 0,81 gewählt. Diese Wahl beruht auf einem automatischen Verfahren zur Bestimmung des optimalen Schwellwertes für FCR das, in [20] ausführlich beschrieben wird. Dabei werden auf der Basis der Trainingsdaten des FCR Systems zwei Dichtefunktionen berechnet, von denen die eine die Verteilung der Qualitätsmaße der falsch klassifizierten Kontexte, die andere die Verteilung der richtig klassi-

¹Der eingesetzte AwarePen ist mit einem FFNN mit 7 versteckten Neuronen in der inneren Schicht ausgestattet.

fizierten Kontexte beschreibt. Ausgehend von diesen Dichten wird dann mittels eines ML-Schätzers ein Schwellwert so berechnet, dass die Wahrscheinlichkeit sowohl für fälschlicherweise abgelehnte Kontexte (engl. false negatives, FN), als auch für fälschlicherweise akzeptierte Kontexte (engl. false positives) minimal wird.

Ges.	Erkannt	E-Rate	Abgel.	FN	E-Rate FCR
1062	863	81,26%	212	53	95,29%
322	119	36,96%	194	14	82,03%
642	426	66,36%	224	34	93,78%
628	340	54,14%	318	48	94,19%
496	364	73,39%	112	28	87,50%
564	334	59,22%	232	52	84,94%
946	665	70,30%	194	30	84,41%
4660	3111	66,76%	1486	259	89,86%

Tabelle 6.2. Feldversuch FCR: Ergebnisse

Tabelle 6.2 zeigt die Ergebnisse der Versuche. Jede der Zeilen enthält die Ergebnisse einer Versuchsperson. In der letzten Zeile sind die Ergebnisse aller Versuche summiert. Die erste Spalte zeigt die Gesamtzahl der Kontexte, die von dem AwarePen während des Versuchs produziert wurden. In der zweiten Spalte ist die Anzahl der korrekt erkannten Kontexte ohne Qualitätsfilterung angegeben. Die sich aus diesen Zahlen ergebende Erkennungsrate für den Stiftkontext ist in Spalte drei der Tabelle zu finden. Soweit gibt die Tabelle die Versuche ohne aktivierte Filterung auf das Attribut FCR wieder. Die vierte Spalte der Tabelle, welche mit „Abgel.“ überschrieben ist, enthält die Anzahl der Kontexte, welche bei aktiver Filterung mit dem Schwellwert von 0,81 abgelehnt wurden. In der Spalte „FN“ ist die Anzahl der Kontexte notiert, welche fälschlicherweise abgelehnt wurden. Die Erkennungsrate mit aktivierter FCR Filterung ist in der letzten Spalte der Tabelle aufgeführt.

Über alle Versuche konnte durch die FCR Filterung die Erkennungsrate des Stiftkontextes von 66,76% um 22,92% auf 89,86% gesteigert werden. Von 4660 produzierten Kontexten wurde dabei 1486 als qualitativ nicht akzeptabel zurückgewiesen und damit nicht weiterverarbeitet. Die maximal bei diesen Versuchen erzielte Verbesserung lag bei 45,07%.

Auch hier wurde die Verbesserung der Erkennungsrate erreicht, ohne den Erkennungsalgorithmus selbst zu verbessern.

6.3.3 Feldversuch: GRC

Für die Evaluierung des Attributes Verwandtschaft von Kontexten, wurde der Versuchsaufbau aus der simulativen Evaluierung in das AwareOffice Umfeld übertragen. Um das Verhalten des Algorithmus bei der Auffächerung von Kontexten zu beobachten, wurde die Erkennungsrate des interaktiven Türschildes von Besprechungskontexten herangezogen. Für den Versuch wurden drei verschiedene Versionen des PDA Artefaktes eingesetzt. Alle PDA Versionen können in diesem Versuch einen Besprechungskontext erkennen. Allerdings verwenden die drei unterschiedlichen Versionen verschiedene Algorithmen, um den Besprechungskontext abzuleiten. PDAs vom Typ 1 schließen den Besprechungskontext aus der Aktivität von mindestens drei Stühlen im AwareOffice. PDAs vom Typ 2 leiten den Besprechungskontext aus den Kontexten des AwarePen und der Tafel ab. Wird mit dem Stift an der Tafel geschrieben, produzieren diese PDAs einen Besprechungskontext. Die PDAs des dritten Typs produzieren einen Besprechungskontext immer dann, wenn die Tür des AwareOffice geschlossen ist. Das interaktive Türschild ist so programmiert, dass es seinen Besprechungskontext aus den Besprechungskontexten der PDAs im AwareOffice durch Fusion ableitet. Für den Versuch wurden fünf PDA Artefakte verwendet. Das Türschild erkennt einen Besprechungskontext, wenn mindestens drei dieser PDAs eine Besprechung erkennen.

Um die Auswirkung der Auffächerung von Kontexten zu untersuchen, wurden für diesen Versuch ein PDA vom Typ 1, ein PDA vom Typ 2 und drei PDAs vom Typ 3 im AwareOffice aktiviert. In dieser Konfiguration kann ohne Qualitätsfilterung eine Fehlerkennung eintreten, wenn weniger als drei Stühle besetzt sind und nicht an der Tafel geschrieben wird, die Tür aber geschlossen ist, da in diesem Fall drei der Artefakte einen Besprechungskontext produzieren, dieser aber auf nur eine einzige Informationsquelle, die geschlossene Tür, zurückgeht.

Der Ablauf dieses Versuchs bestand aus einer Besprechung mit geschlossener Tür und Tafelanschrieb, in deren Anschluss die Teilnehmer das AwareOffice für eine Pause verlassen und die PDAs auf dem Tisch zurückbleiben. Die Tür wird nach dem Verlassen des Raumes geschlossen.

Bei deaktivierter GRC Filterung wurde die Besprechung vom Türschild mit einer durchschnittlichen Erkennungsrate von 0,87 erkannt. Im zweiten Versuchsteil wurde allerdings in 96% der Fälle fälschlich auch eine Besprechung erkannt. Diese Erkennungsrate entspricht der des Tür-Artefaktes. Die Erkennungsrate für den gesamten Versuch lag somit bei 0,455.

Für den Vergleichsversuch mit aktiver Qualitätsfilterung, wurde der Qualitätsfilter des interaktiven Türschildes so konfiguriert, dass er Kontexte, die mindestens 50% ihrer Information gemeinsam haben, also einen Verwandtschaftsgrad größer 0,5 als aufweisen, ablehnt. Die Erkennungsrate im ersten Versuchsteil blieb dabei unverändert bei durchschnittlich 0,87. Im zweiten Versuchsteil kam es dagegen zu keinen Fehlerkennungen. Bei allen zu konsumierenden Besprechungskontexten der PDAs konnte das Türschild anhand des Verwandtschaftsgrades erkennen, dass diese Kontexte auf die gleiche Quelle zurückgehen. Die Gesamterkennungsrate für diesen Versuch lag damit bei 0,935.

Die Evaluierung der Leistung von GRC bei der zyklischen Verarbeitung von Kontexten wurde ebenfalls auf der Basis der Erkennungsrate des Türschildes für Besprechungskontexte durchgeführt. Die Anwendung der PDAs wurde dahingehend modifiziert, dass sie eine Besprechung erkennt, wenn mindestens drei der Stühle aktiv sind, oder ein anderer Besprechungskontext verfügbar ist. Die Funktionalität des Türschildes wurde so reduziert, dass lediglich eine Besprechung erkannt werden kann, wenn ein Besprechungskontext konsumiert wird. Auch der Versuchsablauf blieb weitgehend unverändert: Nach einer Besprechung mit drei Personen verlassen diese das AwareOffice zu einer Kaffeepause. Ein PDA bleibt dabei im Raum zurück.

Ohne Filterung des Attributes GRC lag die Erkennungsrate des Türschildes im ersten Versuchsteil durchschnittlich bei 0,81. Nachdem die Teilnehmer das AwareOffice verlassen hatten, wurde aber weiterhin eine Besprechung erkannt. Dies ist darauf zurück zu führen, dass sowohl der vergessene PDA, als auch das Türschild Besprechungskontexte konsumieren und produzieren.

Bei aktivierter Filterung von einem Verwandtschaftsgrad größer als 0,5 konnte sich der Verarbeitungszyklus im Vergleichsversuch nicht ausbilden. Der Abgleich des Genoms des zu konsumierenden Besprechungskontextes mit den gespeicherten Genomen im Historienspeicher des Artefaktes verhindert dies. Damit lag in diesem Versuch auch die Gesamterkennungsrate bei einem Wert von 0,81.

6.4 Zusammenfassung

Die Evaluierung des, in dieser Arbeit entwickelten und umgesetzten Qualitätsmanagementsystems für Kontexte in ubiquitären Informationsumgebungen zeigt, dass sowohl in der Simulation, als auch in der Anwendung in realen Versuchsumgebungen durch den Einsatz von Kontextattributen zur Filterung von Kontexten, die Erkennungsrate kontextsensitiver Anwendungen signifikant verbessert werden kann. Von den Verbesserungen der Kontexterkennungsraten durch die Einführung einer Kontextmanagementschicht zur Sicherung der Kontextqualität, profitieren alle kontextsensitiven Anwendungen unabhängig von der Qualität ihrer Erkennungsalgorithmen.

In diesem Kapitel konnte nachgewiesen werden, dass das hier evaluierte System zu einer realen, quantitativ messbaren Verbesserung der Kontexterkennungsrate von kontextsensitiven Anwendungen führt. Der Hauptunterschied zu früheren Arbeiten zum Thema Qualitätssicherung von Kontexten ist, dass hier ein konkretes System entwickelt werden konnte, dass die ihm zu Grunde liegenden theoretischen Überlegungen zu Qualitätssicherungsmaßnahmen konsequent umsetzt und somit anwendbar macht.

Die Simulation der verschiedenen Szenarien hat gezeigt, dass in bestimmten Fehlersituationen die Erkennungsraten von Artefakten, durch das Kontextmanagementsystem um über 50% verbessert werden können. Die durchschnittliche Erkennungsrate aller Artefakte einer simulierten ubiquitären Informationsumgebung konnte durch die Anwendung aller vier Kontextattribute von 53,99% auf 80,19% verbessert werden.

Die Evaluierung der realen Implementierung des Kontextmanagementsystems auf den Artefakten des AwareOffice konnte diese Ergebnisse bestätigen und unterstreichen. Bereits heutige Installationen von ubiquitären Informationsumgebungen, vergleichsweise geringer Komplexität und Dynamik, können signifikant von dem vorgestellten System profitieren. Durch die Anwendung des Attributs CA konnten Verbesserungen der Erkennungsrate von bis zu 33% erreicht werden. Die durchschnittlich erzielbare Verbesserung durch das Attribut LOC lag im gleichen Bereich. Die maximale Verbesserung durch das Attribut FCR lag mit 45,07% dagegen noch einmal deutlich höher. Für die Anwendung des Attributs GRC konnte sogar eine Verbesserung von bis zu 48% nachgewiesen werden.

7 Zusammenfassung und Ausblick

Diese Arbeit führte ein neues System zur Sicherung der Qualität von Kontextdaten in ubiquitären Informationsumgebungen ein. Das zentrale Ziel der Entwicklung dieses Kontextmanagementsystems war die Verbesserung der Kontexterkennungsrate der Artefakte einer Umgebung. Die Grundidee der hier präsentierten Lösung war die funktionale Modellierung der Kontextverarbeitung und -kommunikation unter Einbindung einer gesonderten Kontextmanagementschicht, welche die Aufgaben der Qualitätssicherung wahrnimmt. Basierend auf diesem Modell wurde die Architektur einer ubiquitären Informationsumgebung mit eingebetteter Kontextqualitätsfunktionalität entwickelt. Diese wurde prototypisch umgesetzt und mit der Methodik des „Living Lab“ in der Praxis evaluiert. Ergänzend wurde eine simulative Evaluierung des Systems durchgeführt, welche Anwendungsszenarien abdeckt, die mit heutigen technischen Mitteln noch nicht real umgesetzt werden können.

7.1 Zusammenfassung der Ergebnisse

Vergleicht man existierende Ansätze zur Qualitätssicherung in kontextsensitiven Systemen, ist das hier vorgestellte Kontextmanagementsystem das einzige, das eine konkrete Lösung vorschlägt, welche die quantitative Bewertung der Kontextqualität (engl. Quality of Context, QoC) erlaubt. Die Effektivität dieses Qualitätssicherungssystems konnte durch Simulation, Implementierung und Feldversuche nachgewiesen werden. Die Vorteile der Lösung sind:

- **Dezentrale Funktion:** Das Kontextmanagementsystem ist vollständig dezentral aufgebaut und bindet sich so optimal in die Strukturen ubiquitärer Informationsumgebungen ein. Alle eingesetzten Algorithmen arbeiten ausschließlich auf der Basis lokalen Wissens, das dem kontextverarbeitenden Artefakt zur Verfügung steht. Globales Wissen über die Verarbeitungs- und Kommunikationsstrukturen der konkreten Umgebung sind für die Funktion nicht notwendig.

- **Unabhängigkeit von Kontextalgorithmen:** Die Qualitätssicherung ist als unabhängige Schicht innerhalb der Artefaktarchitektur realisiert. Dies ermöglicht die vollständige Entkoppelung der Kontexterkennung von der Qualitätssicherung. Bestehende Kontextalgorithmen müssen somit nicht an das Kontextmanagementsystem angepasst werden. Dennoch können diese ohne Veränderungen von der Qualitätssicherung durch höhere Erkennungsraten profitieren.
- **Effiziente Funktion:** Das System ist effizient, da Modell und Architektur für den Einsatz in ubiquitären Informationsumgebungen optimiert wurden. Darüber hinaus berücksichtigt die Implementierung praktische Rahmenbedingungen. Durch Simulation und konkrete Feldversuche konnte die Effizienz des Kontextmanagementsystems nachgewiesen werden.

Wachsende und damit komplexer werdende ubiquitäre Informationsumgebungen stellen den Entwickler von kontextsensitiven Anwendungen vor neue Herausforderungen. Mit einer steigenden Anzahl von, über den Austausch von Kontextinformationen, kommunizierenden Artefakten und der damit einhergehenden Weiterentwicklung in der Verarbeitung höherer Kontexte, steigen auch die Anforderungen an die Qualität dieser Kontexte.

Um eine Verbesserung der Erkennungsraten von Kontexten jenseits der Optimierung der Erkennungsalgorithmen zu erreichen, wurde ein Modell zur einheitlichen Repräsentation, Kommunikation und Verarbeitung von Kontexten entwickelt. Dieses Modell berücksichtigt sowohl Eigenschaften der Anwendungsdomäne, als auch technische Rahmenbedingungen. Zentraler Bestandteil ist das Modell attributierter Kontexte. Dieses ordnet allen Kontexten einer ubiquitären Informationsumgebung, unabhängig von ihrer semantischen Bedeutung, funktionale Eigenschaften zu. Die Kontexteigenschaften, oder Attribute, haben allgemeinen beschreibenden Charakter. Sie sind so ausgewählt, dass sie auf alle Kontexte zutreffen und diese in einer Art und Weise beschreiben, die es kontextsensitiven Anwendungen erlaubt, über die Auswertung dieser Attribute, Rückschlüsse auf die Qualität der Kontexte zu ziehen.

Neben den bereits früher in der Literatur beschriebenen Einflussfaktoren auf die Kontextqualität, das Alter des Kontextes und den Ort der Kontextentstehung, werden die Verlässlichkeit des eingesetzten Kontextalgorithmus und die informationelle Abhängigkeit von Kontextdaten als Kontextattribute eingeführt. Alle vier Attribute werden durch die Architektur des Kontextmanagementsystems in dieser

Arbeit zum ersten Mal mit konkreten Algorithmen hinterlegt, welche die quantitative Berechnung von Qualitätsmaßen auf der Basis dieser Attribute erlaubt. Der Qualitätsfilter gestattet es Artefakten, Kontexte aufgrund dieser Maße, den Anforderungen der Anwendung entsprechend, zu filtern.

Das zentrale Ziel dieser Arbeit war die Entwicklung eines Systems, welches die Erkennungsrate von Kontexten in ubiquitären Informationsumgebungen signifikant verbessern kann. Die simulative und praktische Evaluierung des vorgeschlagenen Kontextmanagementsystems zeigt, dass dieses Ziel erreicht werden konnte. In der Simulation konnte beobachtet werden, dass durch den Einsatz des Qualitätsmanagementsystems die durchschnittliche Kontexterkennungsrate einer ubiquitären Informationsumgebung von 53,99% auf 80,19% verbessert werden konnte. Dabei wurden 98 Artefakte und über 634.000 Kontextverarbeitungen simuliert. In speziellen Problemszenarien konnte sogar eine Verbesserung der Erkennungsrate um bis zu 57,03% nachgewiesen werden. Die praktische Evaluierung konnte die simulativen Ergebnisse bestätigen. Hier konnten zum Beispiel Verbesserungen von bis zu 45,07% durch das Attribut *Verlässlichkeit von Kontexten* (*Fuzzy Context Reliability, FCR*) und Verbesserungen von bis zu 48,0% durch das Attribut *Verwandschaft von Kontexten* (*Genetic Relation of Contexts, GRC*) nachgewiesen werden.

7.2 Ausblick

Das Kontextmanagementsystem bildet den ersten Schritt in der funktionalen Verarbeitung von Kontexten. Diese Arbeit liefert erste Beiträge und Vorschläge zur Vereinheitlichung von Repräsentation, Kommunikation und Verarbeitung von Kontexten. Dieser zentrale Bereich der Kontextverarbeitung wird in naher Zukunft weiter an Bedeutung gewinnen, da sich die Technologie und die Anwendungen für ubiquitäre Informationsumgebungen schnell entwickeln und bereits beginnen Marktreife zu erreichen. Die Modelle dieser Arbeit können hier Anknüpfungspunkte für weitere Forschungsarbeiten liefern.

Im Bereich der Qualitätssicherung von Kontexten erschließt die vorliegende Arbeit Neuland. Zum ersten Mal wird hier ein System vorgestellt, welches konkrete Qualitätsmaße für Kontexte vorschlägt und diese auch algorithmisch umsetzt. Eine interessante Forschungsfrage die sich hier ergibt, ist zum Beispiel, in wieweit die vorgestellten Algorithmen auf andere Bereiche, die sich ebenfalls mit der Optimierung von algorithmischen Erkennern beschäftigen, übertragbar sind.

Eine weiterführende Untersuchung des Potentials, der für FCR eingesetzten Algorithmen, könnte Impulse auch für die Forschung an semantischen Kontextalgorithmen generieren. Das Konzept automatisch generierbarer Erkennen hat das Potential die Entwicklung neuer kontextsensitiver Anwendungen erheblich zu vereinfachen und zu beschleunigen.

Die genetischen Konzepte hinter GRC eröffnen die Möglichkeit mit minimalem Ressourceneinsatz informationelle Abhängigkeiten von Daten in dezentralen Strukturen zu dokumentieren und evaluierbar zu machen. Hier können in weiteren Forschungsarbeiten ebenfalls neue Anwendungsfelder, zum Beispiel in verteilten Systemen und für Sensornetzwerke, erschlossen werden.

Konkrete Weiterentwicklungen des in dieser Arbeit vorgestellten Kontextmanagementsystems können die Algorithmen selbst betreffen. Das Zusammenspiel von Clusteringverfahren, ANFIS (Adaptive-network-based Fuzzy Inference System) Identifikation und Parameteroptimierung kann sicher noch weiter verbessert werden. Interessant wäre zum Beispiel ein Vergleich der Leistungsfähigkeit des FCR Systems unter der Verwendung anderer als der vorgeschlagenen Umsetzung der unscharfen Operatoren. Soll das Verfahren auf andere Anwendungsbereiche übertragen werden, muss auch untersucht werden, ob sich dort das subtraktive Clustering weiterhin für die initiale Strukturbestimmung eignet.

Die Verwandtschaftsmatrizen, welche bei der Verarbeitung mehrerer Kontexte aufgestellt werden, wurden im Rahmen dieser Arbeit nur verwendet, um paarweise den Verwandtschaftsgrad von Kontexten zu evaluieren. Es ist zu erwarten, dass aus der gesamten Matrix weitere interessante Informationen extrahiert werden können, welche zu einer weiteren Optimierung der Filterung durch den Qualitätsfilter verwendet werden können.

Die Forschung im Bereich QoC steht immer noch am Anfang. Diese Arbeit schlägt einen ersten konkreten Weg zur Lösung der in wachsenden ubiquitären Informationsumgebungen entstehenden Probleme vor. Es ist zu wünschen, dass sie andere Forscher inspirieren kann, weitere kreative Arbeiten in diesem Bereich anzugehen.

Literaturverzeichnis

- [1] *International Standard ISO/IEC 7498-1 Information technology - Open-systems Interconnection - Basic Reference Model: The Basic Model*
- [2] *GOLD Genomes OnLine Database v2.0*. <http://www.genomesonline.org/>. Aug. 2006
- [3] *SeSAm - Multi-Agent Simulation Environment*. <http://www.simsesam.de>. 2006
- [4] *SWARM*. <http://www.swarm.org>. 2006
- [5] *The Network Simulator - ns2*. <http://nsnam.isi.edu/nsnam/>. 2006
- [6] ABOWD, G. D.: Classroom 200: An Experiment with the Instrumentation of a Living Education Environment. In: *IBM Systems Journal, Special Issue on Pervasive Computing* 38(4) (1999), S. 508–530
- [7] ABOWD, Gregory D. ; MYNATT, Elizabeth D.: Charting past, present, and future research in ubiquitous computing. In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 7(1) (2000), S. 29 – 58
- [8] BAO, Ling ; INTILLE, Stephen S.: Activity Recognition from User-Annotated Acceleration Data. In: *Pervasive*, 2004, S. 1–17
- [9] BARDRAM, Jakob E.: The Java Context Awareness Framework (JCAF) - A Service Infrastructure and Programming Framework for Context-Aware Applications. In: *Proceedings of the 3rd International Conference on Pervasive Computing (Pervasive 2005)*, 2005
- [10] BARTON, John ; VIJAYARAGHAVAN, Vikram: UBIWISE, A Ubiquitous Wireless Infrastructure Simulation Environment. In: *UbiComp 2002*, 2002
- [11] BARTON, John ; VIJAYARAGHAVAN, Vikram: UBIWISE, A Simulator for Ubiquitous Computing Systems Design / Hewlett-Packard Development Company. 2003. – Forschungsbericht

- [12] BEDWORTH, M.D.: Markov chain prediction fusion for automatic target recognition. In: *First Australian Data Fusion Symposium*, 1996
- [13] BEIGL, Michael: Using spatial Co-location for Coordination in Ubiquitous Computing Environments. In: GELLERSEN, H.-W. (Hrsg.): *1th International Symposium on Handheld and Ubiquitous Computing (HUC99)* Bd. 1707. Karlsruhe, Germany : Springer, 1999, S. 259–273
- [14] BEIGL, Michael: *Kommunikation in interaktiven Räumen*, Universität Karlsruhe, Diss., 2001
- [15] BEIGL, Michael ; GELLERSEN, H.: Smart-Its: An Embedded Platform for Smart Objects. In: *Smart Objects Conference (sOc)*, 2003
- [16] BEIGL, Michael ; GELLERSEN, Hans-Werner ; SCHMIDT, Albrecht: MediaCups: Experience with Design and Use of Computer-Augmented Everyday Objects. In: *Computer Networks, Special Issue on Pervasive Computing* 35 (2001), March, Nr. 4, S. 401–409
- [17] BEIGL, Michael ; KROHN, Albert ; ZIMMER, Tobias ; DECKER, Christian ; ROBINSON, Philip: AwareCon: Situation Aware Context Communication. In: *Proceedings of Ubicomp 2003*. Seattle, USA, Oktober 2003
- [18] BEIGL, Michael ; ZIMMER, Tobias ; DECKER, Christian: A location model for communicating and processing of context. In: *Personal and Ubiquitous Computing Journal* 6 (2002), Nr. 5-6, S. 341–357
- [19] BEIGL, Michael ; ZIMMER, Tobias ; KROHN, Albert ; DECKER, Christian ; ROBINSON, Philip: Smart-Its – Communication and Sensing Technology for UbiComp Environments / University of Karlsruhe. 2003 (2003/2). – ISSN 1432-7864
- [20] BERCHTHOLD, Martin: *Qualitätsmanagement Von Kontexten - Verlässlichkeit in Fuzzy Logic*, Universität Karlsruhe (TH), Diplomarbeit, 2006
- [21] BERNERS-LEE, Tim ; HENDLER, James ; LASSILA, Ora: The Semantic Web. In: *Scientific American* May (2001)
- [22] BHARATULA, Nagendra B. ; STÄGER, Mathias ; LUKOWICZ, Paul ; TRÖSTER, Gerhard: Empirical Study of Design Choices in Multi-Sensor Context Recognition Systems. In: *IFAWC: 2nd International Forum on Applied Wearable Computing*, 2005, S. 79–93

- [23] BROWN, Peter J. ; BOVEY, John D. ; CHEN, Xian: Context-aware applications: From the laboratory to the marketplace. In: *IEEE Personal Communications* 4(5) (1997), October, S. 58–64
- [24] BUCHHOLZ, Thomas ; KÜPPER, AxelL ; SCHIFFERS, Michael: Quality of Context: What it is and why we need it. In: *10th Workshop of the HP OpenView University Association*, 2003
- [25] BUSSMANN, Hadumod ; KRÖNER (Hrsg.): *Lexikon der Sprachwissenschaft*. Stuttgart : Hadumod Bußmann, 2002
- [26] BUXTON, W.: Ubiquitous Media and the Active Office. In: *Nikkei Electronics* 3.27 (1995), Nr. 632, S. 187–195
- [27] CAO, Jiannong ; XING, Na ; CHAN, Alvin T. S. ; FENG, Yulin ; JIN, Beihong: Service Adaptation Using Fuzzy Theory in Context-Aware Mobile Computing Middleware. In: *rtcsa* 0 (2005), S. 496–501. – ISSN 1533–2306
- [28] CASTRO, P. ; MUNTZ, R.: Managing context for smart spaces. In: *IEEE Personal Communications* vol.7, no. 5 (2000), October, S. 44–46
- [29] CASTRO, Paul ; CHIU, Patrick ; KREMENEK, Ted ; MUNTZ, Richard R.: A Probabilistic Room Location Service for Wireless Networked Environments. In: *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing*. London, UK : Springer-Verlag, 2001. – ISBN 3–540–42614–0, S. 18–34
- [30] CHEN, Guanling: *Solar: Building A Context Fusion Network for Pervasive Computing*, DARTMOUTH COLLEGE Hanover, New Hampshire, Diss., 2004
- [31] CHEN, Harry: *An Intelligent Broker Architecture for Pervasive Context-Aware Systems*, University of Maryland, Diss., December 2004
- [32] CHEN, Harry ; FININ, Tim ; JOSHI, Anupam: An ontology for context-aware pervasive computing environments. In: *Knowl. Eng. Rev.* 18 (2003), Nr. 3, S. 197–207. – ISSN 0269–8889
- [33] CHEN, J.-G. ; ANSARI, N: Adaptive fusion of correlated local decisions. In: *IEEE Transactions on Systems, Man and Cybernetics* vol.28, no.2 (1998), S. 276–281

- [34] CHIU, S: Method and software for extracting fuzzy classification rules by subtractive clustering. In: *Fuzzy Information Processing Society NAFIPS. Biennial Conference of the North American*, 1996
- [35] CHIU, S. L.: Fuzzy model identification based on cluster estimation. In: *Journal of Intelligent and Fuzzy Systems* 2(3) (1994), S. 267–277
- [36] CHIU, Stephen: Extracting Fuzzy Rules for Pattern Classification by Cluster Estimation. In: *6'th Int. Fuzzy Systems Association Congress (IFSA '95)*, 1995
- [37] CHIU, Stephen: An Efficient Method for Extracting Fuzzy Classification Rules from High Dimensional Data. In: *Journal of Advanced Computational Intelligence* 1(1) (1997), S. 31–36
- [38] CHOW, C. K. ; LIU, C. N.: Approximating discrete probability distributions with dependence trees. In: *IEEE Transactions of Information Theory* 14(3) (1968), S. 462–467
- [39] CHRISTOPOULOU, E. ; KAMEAS, A.: Using ontologies to address key issues in ubiquitous computing systems. In: *proceedings of the 2nd European Symposium on Ambient Intelligence (EUSAI)* Bd. LNCS 3296, 2004, S. 13–24
- [40] CHRISTOPOULOU, Eleni ; GOUMOPOULOS, Christos ; KAMEAS, Achilles: An ontology-based context management and reasoning process for UbiComp applications. In: *sOc-EUSAI '05: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence*. New York, NY, USA : ACM Press, 2005. – ISBN 1–59593–304–2, S. 265–270
- [41] CHRISTOPOULOU, Eleni ; KAMEAS, Achilles: GAS ontology: an ontology for collaboration among ubiquitous computing devices. In: *Int. J. Hum.-Comput. Stud.* 62 (2005), Nr. 5, S. 664–685. – ISSN 1071–5819
- [42] CLARKSON, B. ; PENTLAND, A.: Extracting context from environmental audio. In: *2nd International Symposium on Wearable Computers*, 1998
- [43] COOPERSTOCK, Jeremy R. ; TANIKOSHI, Koichiro ; BEIRNE, Garry ; NARINE, Tracy ; BUXTON, William A. S.: Evolution of a reactive environment. In: *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM Press/Addison-Wesley Publishing Co., 1995. – ISBN 0–201–84705–1, S. 170–177

- [44] COROAMA, Vlad ; HÄHNER, Jörg ; HANDY, Matthias ; RUDOLPH-KUHN, Patricia ; MAGERKURTH, Carsten ; MÜLLER, Jürgen ; STRASSER, Moritz ; ZIMMER, Tobias: Leben in einer smarten Umgebung - Ubiquitous Computing: Szenarien und Auswirkungen. In: *Gottlieb Daimler- und Karl Benz-Stiftung* (2003), 12
- [45] COUTAZ, Joelle ; CROWLEY, James L. ; DOBSON, Simon ; GARLAN, David: Context is key. In: *Communications of the ACM* 48 (2005), Nr. 3, S. 49–53. – ISSN 0001–0782
- [46] DECKER, Christian ; KROHN, Albert ; BEIGL, Michael ; ZIMMER, Tobias: The Particle Computer System. In: *Proceedings of the Fourth International Conference IPSN*. Los Angeles, USA, 2005
- [47] DECKER, Christian ; SPIESS, Patrik ; SOUZA, Luciana M. d. ; BEIGL, Michael ; NOCHTA, Zoltan: Coupling Enterprise Systems with Wireless Sensor Nodes: Analysis, Implementation, Experiences and Guidelines. In: *Pervasive Technology Applied (PTA), Workshop on Pervasive 2006*, 2006
- [48] DEY, A. K.: Understanding and using context. In: *Personal and Ubiquitous Computing Personal and Ubiquitous Computing, Special issue on Situated Interaction and Ubiquitous Computing* 5, 1 (2001), S. 4–7
- [49] DEY, A. K. ; ABOWD, G. D. ; SALBER, D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. In: *Human-Computer Interaction* 16 (2-4) (2001), S. 97–166
- [50] DEY, A.K.: *Providing Architectural Support for Building Context-Aware Applications*, Georgia Institute of Technology, Diss., December 2000
- [51] DEY, A.K. ; MANKOFF, J. ; ABOWD, G.D: Distributed Mediation of Imperfectly Sensed Context in Aware Environments / GVU Technical Report No. 00-14. 2000. – Forschungsbericht
- [52] DEY, Anind K. ; MANKOFF, Jennifer: Designing mediation for context-aware applications. In: *ACM Trans. Comput.-Hum. Interact.* 12 (2005), Nr. 1, S. 53–80. – ISSN 1073–0516
- [53] DEY, Anind K. ; MANKOFF, Jennifer ; ABOWD, Gregory D. ; CARTER, Scott: Distributed Mediation of Ambiguous Context in Aware Environments. In: *Proceedings of the 15th Annual Symposium on User Interface*

- Software and Technology (UIST 2002)*. Paris, France, Oktober 2002, S. 121–130
- [54] DUCATEL, K. ; BOGDANOWICZ, M. ; SCAPOLO, F. ; LEIJTEN, J. ; BURGELMAN, J-C. ; ISTAG (Hrsg.): *SCENARIOS FOR AMBIENT INTELLIGENCE IN 2010*. The IST Advisory Group European Commission, 2001
- [55] EBLING, M. ; HUNT, G. ; LEI, H.: Issues for Context Services for Pervasive Computing. In: *Middleware'01 Advanced Workshop on Middleware for Mobile Computing*, 2001
- [56] EDWARDS, W. K. ; GRINTER, Rebecca E.: At Home with Ubiquitous Computing: Seven Challenges. In: *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing*. London, UK : Springer-Verlag, 2001. – ISBN 3–540–42614–0, S. 256–272
- [57] ETTER, R. ; DOCKHORN COSTA, P. ; BROENS: A Rule-Based Approach Towards Context-Aware User Notification Services. In: *IEEE International Conference on Pervasive Services*, 2006
- [58] FRANKLIN, David ; FLASCHBART, Joshua: All gadget and no representation makes jack a dull environment. In: *AAAI 1998 Spring Symposium on Intelligent Environments*. Palo Alto, CA : AAAI Press, March 1998, S. 155–160
- [59] FRIEDMAN, Nir ; GEIGER, Dan ; GOLDSZMIDT, Moises: Bayesian Network Classifiers. In: *Machine Learning* 29 (1997), Nr. 2-3, S. 131–163
- [60] GELERENTER, David ; BERNSTEIN, Arthur J.: Distributed communication via global buffer. In: *PODC '82: Proceedings of the first ACM SIGACT-SIGOPS symposium on Principles of distributed computing*. New York, NY, USA : ACM Press, 1982. – ISBN 0–89791–081–8, S. 10–18
- [61] GLASSEY, R. ; STEVENSON, G. ; RICHMOND, M. ; NIXON, P. ; TERZIS, S. ; WANG, F. ; FERGUSON, R. I.: Towards a Middleware for Generalised Context Management. In: *First International Workshop on Middleware for Pervasive and Ad Hoc Computing, Middleware 2003*, 2003
- [62] GOLDBERG, David E. ; GOLDBERG, David E. (Hrsg.): *Genetic Algorithms in Search, Optimazation, and Machine Learning*. Addison Wesley, 1989

- [63] GRAY, Philip D. ; SALBER, Daniel: Modelling and Using Sensed Context Information in the Design of Interactive Applications. In: *EHCI '01: Proceedings of the 8th IFIP International Conference on Engineering for Human-Computer Interaction*. London, UK : Springer-Verlag, 2001. – ISBN 3–540–43044–X, S. 317–336
- [64] GREENBERG, Saul: Context as a Dynamic Construct. In: *Human-Computer Interaction* 16 (2-4) (2001), S. 257–268
- [65] GU, T. ; PUNG, H. K. ; ZHANG, D. Q.: A Bayesian Approach for Dealing with Uncertain Contexts. In: KOTSIS, G. (Hrsg.): *Proceedings of the 2nd International Conference on Pervasive Computing*, Austrian Computer Society, 2004
- [66] GU, T. ; PUNG, H. K. ; ZHANG, D. Q.: A Middleware for Building Context-Aware Mobile Services. In: *Proceedings of IEEE Vehicular Technology Conference (VTC 2004)*, 2004
- [67] GU, T. ; WANG, X. ; PUNG, H. ; ZHANG, D.: An Ontology-based Context Model in Intelligent Environments. In: *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference*, 2004
- [68] HANSS, Michael: *Identifikation von Fuzzy-Modellen und ihre Anwendung zur Regelung nichtlinearer Prozesse*, Universität Stuttgart, Diss., 1998
- [69] HELD, A. ; BUCHHOLZ, S. ; SCHILL, A.: Modeling of Context Information for Pervasive Computing Applications. In: *Proc. of the 6th World Multiconference on Systemics, Cybernetics and Informatics*, 2002
- [70] HELLENDORRN, Hans (Hrsg.) ; DRIANKOV, Dimiter (Hrsg.): *Fuzzy model identification: selected approaches*. London, UK : Springer-Verlag, 1997. – ISBN 3–540–62721–9
- [71] HENRICKSEN, K. ; INDULSKA, J.: A software engineering framework for context-aware pervasive computing. In: *Pervasive Computing and Communications (PerCom)*, 2004
- [72] HENRICKSEN, K. ; INDULSKA, J.: Developing context-aware pervasive computing applications: Models and approach. In: *Journal of Pervasive and Mobile Computing* 2(1) (2006), February, S. 37–64

- [73] HENRICKSEN, K. ; INDULSKA, J. ; RAKOTONIRAINY, A.: Modeling context information in pervasive computing systems. In: MATTERN, F. (Hrsg.) ; NAGHSHINEH, M. (Hrsg.): *Proceedings of 1st International Conference on Pervasive Computing*. Zurich, Switzerland : Springer, 2002, S. pp. 167–180
- [74] HENRICKSEN, Karen ; INDULSKA, Jadwiga: Modelling and Using Imperfect Context Information. In: *percomw 00* (2004), S. 33. ISBN 0-7695-2106-1
- [75] HENRICKSEN, Karen ; INDULSKA, Jadwiga ; MCFADDEN, Ted ; BALASUBRAMANIAM, Sasitharan: Middleware for Distributed Context-Aware Systems. In: *Lecture Notes in Computer Science* Bd. 3760. Springer, Oct 2005, S. 846 – 863
- [76] HENRICKSEN, Karen ; INDULSKA, Jadwiga ; RAKOTONIRAINY, Andry: Generating Context Management Infrastructure from Context Models. In: *4th International Conference on Mobile Data Management (MDM), Industrial Track Proceedings*. Melbourne, January 2003, S. 1–6
- [77] HILTY, L.M. ; TA-SWISS (Hrsg.): *Das Vorsorgeprinzip in der Informationsgesellschaft: Auswirkungen des Pervasive Computing auf Gesundheit und Umwelt*. Swiss Centre for Technology Assessment, 2003
- [78] HINRICKSEN, Karen: *A Framework for Context-Aware Pervasive Computing Applications*, University of Queensland, Diss., 2003
- [79] HÖNLE, Nicola ; KÄPPELER, Uwe-Philipp ; NICKLAS, Daniela ; SCHWARZ, Thomas ; GROSSMANN, Matthias: Benefits of Integrating Meta Data into a Context Model. In: *Proceedings of the 3rd International Conference on Pervasive Computing and Communications*, 2005
- [80] HOLLAND, John H.: *Adaptation in natural and artificial systems*. Cambridge, MA, USA : MIT Press, 1992. – ISBN 0-262-58111-6
- [81] HONG, Jason I. ; LANDAY, James A.: A Context/Communication Information Agent. In: *Personal and Ubiquitous Computing: Special Issue on Situated Interaction and Context-Aware Computing* 5(1) (2001), S. 78–81
- [82] HONG, Jason I. ; LANDAY, James A.: An Infrastructure Approach to Context-Aware Computing. In: *Human-Computer Interaction* 16 (2001), S. 287–303

- [83] HONG, J.I.: The Context Fabric: An Infrastructure for Context-Aware Computing. In: *Human Factors in Computing Systems: CHI*, 2002
- [84] HONG, X. ; XU, K. ; GERLA, M.: Scalable routing protocols for mobile ad hoc networks. In: *IEEE Network Magazine* 16(4) (2002), S. 11–21
- [85] INDULSKA, Jadwiga ; ROBINSON, Ricky ; RAKOTONIRAINY, Andry ; HENRICKSEN, Karen: Experiences in Using CC/PP in Context-Aware Systems. In: *MDM '03: Proceedings of the 4th International Conference on Mobile Data Management*. London, UK : Springer-Verlag, 2003. – ISBN 3–540–00393–2, S. 247–261
- [86] INTILLE, Stephen S.: Change Blind Information Display for Ubiquitous Computing Environments. In: *UbiComp '02: Proceedings of the 4th international conference on Ubiquitous Computing*. London, UK : Springer-Verlag, 2002. – ISBN 3–540–44267–7, S. 91–106
- [87] JANG, J.-S.R.: ANFIS: adaptive-network-based fuzzy inference system. In: *IEEE Transactions on Man and Cybernetics* 23(3) (1993), S. 665–685
- [88] JANG, J.-S.R. ; SUN, Chuen-Tsai: Neuro-fuzzy modeling and control. In: *Proceedings of the IEEE* 83(3) (1995), S. 378–406
- [89] JAYNES, E. T.: Information Theory and Statistical Mechanics. In: *Statistical Physics* 3 (1963), S. 181–218
- [90] JEONG, Jinook ; CHUNG, Seoung: Bootstrap tests for autocorrelation. In: *Computational Statistics & Data Analysis* 38(1) (2001), S. 49 – 69
- [91] JONG, K. A. D.: *An Analysis of the Behavior of a Class of Genetic Adaptive systems*, University of Michigan, Diss., 1975
- [92] KALMAN, R. E.: A New Approach to Linear Filtering and Prediction Problems. In: *Transactions of the ASME - Journal of Basic Engineering* 82 (1960), S. 35–45
- [93] KAM, M. ; ZHU, Q. ; ; GRAY, W.: Optimal data fusion of correlated local decisions in multiple sensor detection systems. In: *IEEE Trans. Aerosp. Electron. Syst.* AES-28 (1992), S. 916–920

- [94] KAMEAS, A. ; BELLIS, S. ; MAVROMMATI, I. ; DELANEY, K. ; COLLEY, M. ; POUNDS-CORNISH, A: An architecture that treats everyday objects as communicating tangible components. In: *Pervasive Computing and Communications*, IEEE, March 2003, S. 115 – 122
- [95] KARMOUCH, A. ; GIAFFREDA, R. ; JONSSON, A. ; KARLSSON, A. ; SMIRNOV, M. ; GLITHO, R. ; GALIS, A.: Context Management Architecture for Ambient Networks. In: *Wireless World Research Forum12*, 2004
- [96] KHEDR, M. ; KARMOUCH, A.: Negotiating context information in context-aware systems. In: *IEEE Intelligent Systems and Their Applications* 19(6) (2004), S. 21– 29
- [97] KIDD, Cory D. ; ORR, Robert J. ; ABOWD, Gregory D. ; ATKESON, Christopher G. ; ESSA, Irfan A. ; MACINTYRE, Blair ; MYNATT, Elizabeth ; STARNER, Thad E. ; NEWSTETTER, Wendy: The Aware Home: A Living Laboratory for Ubiquitous Computing Research. In: *Proceedings of the Second International Workshop on Cooperative Buildings*, 1999
- [98] KIRSH, D.: Adapting the Environment Instead of Oneself. In: *Adaptive Behavior* 4 (3-4) (1996), S. 415–452
- [99] KIRSH, D. ; MAGLIO, P.: On distinguishing epistemic from pragmatic action. In: *Cognitive Science* 18 (1994), S. 513–549
- [100] KIRSH, David: Complementary Strategies: Why we use our hands when we think. In: *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society*, 1995
- [101] KIRSH, David: The intelligent use of space. In: *Artificial Intelligence* 73(1-2) (1995), S. 31–68
- [102] KORPIPÄÄ, Panu ; KOSKINEN, Miika ; PELTOLA, Johannes ; MAKELA,, Satu-Marja ; SEPPÄNEN, Tapio: Bayesian approach to sensor-based context awareness. In: *Personal Ubiquitous Comput.* 7 (2003), Nr. 2, S. 113–124. – ISSN 1617–4909
- [103] KORPIPÄÄ, Panu ; MANTYJARVI, Jani ; KELA, Juha ; KERANEN, Heikki ; MALM, Esko-Juhani: Managing Context Information in Mobile Devices. In: *IEEE Pervasive Computing* 02 (2003), Nr. 3, S. 42–51. – ISSN 1536–1268

- [104] KRAUSE, M. ; HOCHSTÄTTER, I: Challenges in Modelling and Using Quality of Context (QoC). In: *Proceedings of the 2nd International Workshop on Mobility Aware Technologies and Applications*, 2005
- [105] KROGH, A. ; BROWN, M. ; MIAN, I. S. ; SJÖLANDER, K. ; HAUSSLER, D.: Hidden Markov models in computational biology: Applications to protein modeling. In: *Journal of Molecular Biology* 235 (1994), S. 1501–1531
- [106] KROHN, Albert ; BEIGL, Michael ; DECKER, Christian ; ZIMMER, Tobias: ConCom - A language and Protocol for Communication of Context / University of Karlsruhe. 2004 (2004/19). – ISSN 1432-7864
- [107] KUMAR, S. ; RAGHAVAN, V. S. ; DENG, J.: Medium Access Control Protocols for Ad-Hoc Wireless Networks: A Survey. In: *Elsevier Ad-Hoc Networks Journal* 4(3) (2006), S. 326–358
- [108] KUZNETSOV, Igor. *Modulare Integration kontextsensitiver Anwendungen*. Studienarbeit. Mai 2006
- [109] KUZNETSOV, Igor: *Umsetzung und Evaluierung eines Qualitätsmanagementsystems auf Particle Computern*, Universität Karlsruhe (TH), Diplomarbeit, 2006
- [110] LAERHOVEN, K. V.: Combining the Kohonen Self-Organizing Map and K-Means for On-line Classification of Sensordata. In: *Artificial Neural Networks - ICANN 2001*, 2001
- [111] MAMDANI, E. H. ; ASSILIAN, S.: An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. In: *International Journal of Man-Machine Studies* 7 (1) (1975), S. 1–13
- [112] MATT, Julia ; KOHLBECKER, Michael. *Artefakte für das AwareOffice*. Studienarbeit. 2005
- [113] MATTERN, F. ; RÖMER, K.: Drahtlose Sensornetze. In: *Informatik Spektrum* Vol. 26 No. 3 (2003), S. 191–194
- [114] MCCALLUM, Andrew ; JENSEN, David: A Note on the Unification of Information Extraction and Data Mining using Conditional-Probability, Relational Models. In: *SRL2003, IJCAI 2003 Workshop on Learning Statistical Models from Relational Data*, 2003

- [115] MCCARTHY, John: Notes on Formalizing Contexts. In: KEHLER, Tom (Hrsg.) ; ROSENSCHEIN, Stan (Hrsg.): *Proceedings of the Fifth National Conference on Artificial Intelligence*. Los Altos, California : Morgan Kaufmann, 1993, S. 555–560
- [116] MEIR, R. ; ZHANG, T.: Generalization Error Bounds for Bayesian Mixture Algorithms. In: *Journal of Machine Learning Research* 4 (2003), S. 839–860
- [117] MENDES, P. ; PREHOFER, C. ; WEI, Qing: Context management with programmable mobile networks. In: *Computer Communications*, 2003, 2003
- [118] MINAR, N. ; BURKHART, R. ; LANGTON, C. ; ASKENAZI, M.: The Swarm simulation system: A toolkit for building multi-agent simulations / Santa Fe Institute. 1996. – Forschungsbericht
- [119] MÄNNLE, Manfred: *Modellierung nichtlinearer Systeme mit unscharfer Logik*, Universität Karlsruhe (TH), Diplomarbeit, 1995
- [120] MÄNNLE, Manfred: *Unscharfe Modelle zur Systemidentifikation und Fehlererkennung*, Universität Karlsruhe (TH), Diss., 2001
- [121] MORLA, Ricardo ; DAVIES, Nigel: Evaluating a Location-Based Application: A Hybrid Test and Simulation Environment. In: *IEEE Pervasive Computing* 3 (2004), Nr. 3, S. 48–56
- [122] NEVILLE, Jennifer ; RATTIGAN, Matthew ; JENSEN, David: Statistical Relational Learning: Four Claims and a Survey. In: *SRL2003, IJCAI 2003 Workshop on Learning Statistical Models from Relational Data*, 2003
- [123] NORMAN, D.A.: *The Invisible Computer : Why Good Products Can Fail, the Personal Computer Is So Complex, and Information Appliances Are the Solution*. MIT Press, Oktober 1998
- [124] O’NEILL, Eleanor: *TATUS A Ubiquitous Computing Simulator*, University of Dublin, Trinity College, Diplomarbeit, 2004
- [125] O’NEILL, Eleanor ; KLEPAL, Martin ; LEWIS, David ; O’DONNELL, Tony ; O’SULLIVAN, Declan ; PESCH, Dirk: A Testbed for Evaluating Human Interaction with Ubiquitous Computing Environments. In: *TRIDENTCOM*

- '05: *Proceedings of the First International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities (TRIDENTCOM'05)*. Washington, DC, USA : IEEE Computer Society, 2005. – ISBN 0-7695-2219-X, S. 60–69
- [126] PASCOE, Jason: Adding generic contextual capabilities to wearable computers. In: *2nd IEEE International Symposium on Wearable Computers (ISWC'98)*. Pittsburgh, PA, October 1998, S. 92–99
- [127] PIRTTIKANGAS, S. ; RIEKKI, J. ; KAARTINEN, J. ; RÖNING, J.: Context-Recognition and Data Mining Methods for a Health Club Application. In: *12th Int. Conf. Intelligent and Adaptive Systems and Software Engineering (IASSE 2003)*, 2003
- [128] QIN, Liang ; KUNZ, Thomas: Survey on Mobile Ad Hoc Network Routing Protocols and Cross-Layer Design / Carleton University, Systems and Computer Engineering. 2004. – Forschungsbericht
- [129] QUINLAN, J. R.: *C4.5: programs for machine learning*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1993. – ISBN 1-55860-238-0
- [130] RABINER, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. In: *Proceedings of the IEEE* 77(2) (1989), S. 257–286
- [131] RANGANATHAN, A. ; AL-MUHTADI, J. ; CAMPBELL, R.H.: Reasoning about uncertain contexts in pervasive computing environments. In: *Pervasive Computing, IEEE* 3 2 (2004), April-June, S. 62 – 70
- [132] RAYNER, R. K.: Bootstrapping p Values and Power in the First-order Autoregressio: A Monte Carlo Investigation. In: *Business and Economic Statistics* 8 (1990), S. 251–263
- [133] RAZZAQUE, M.A. ; DOBSON, Simon ; NIXON, Paddy: Categorisation and modelling of quality in context information. In: *Proceedings of the IJCAI 2005 Workshop on AI and Autonomic Communications*, 2005
- [134] REIBMAN, A. R. ; NOLTE, L. W.: Design and performance comparison of distributed detection networks. In: *IEEE Trans. Aerosp. Electron. Syst.* AES-23 (1987), S. 789–797

- [135] REKIMOTO, Jun ; AYATSUKA, Yuji ; HAYASHI, Kazuteru: Augment-able reality: Situated communication through physical and digital spaces. In: *2nd IEEE International Symposium on Wearable Computers (ISWC'98)*. Pittsburgh, PA, October 1998, S. 68–75
- [136] REYNOLDS, Vinny ; CAHILL, Vinny ; SENART, Aline: Requirements for an ubiquitous computing simulation and emulation environment. In: *Inter-Sense '06: Proceedings of the first international conference on Integrated internet ad hoc and sensor networks*. New York, NY, USA : ACM Press, 2006. – ISBN 1–59593–427–8, S. 1
- [137] RIEDMILLER, M. ; BRAUN, H.: A direct adaptive method for faster back-propagation learning: The RPROP algorithm. In: *Proceedings International Conference on Neural Networks*, 1993
- [138] ROMAN, M. ; HESS, C. ; CERQUEIRA, R. ; RANGANATHAN, A. ; CAMPBELL, R.H. ; NAHRSTEDT, K.: A middleware infrastructure for active spaces. In: *Pervasive Computing, IEEE* 1(4) (2002), S. 74 – 83
- [139] SADJADI, F. A.: Hypothesis testing in a distributed environment. In: *IEEE Trans. Aerosp. Electron. Syst.* AES-22 (1986), S. 134–137
- [140] SANKOFF, D. ; LEDUC, G. ; ANTOINE, N. ; PAQUIN, B. ; LANG, B. ; CEDERGREN, R.: Gene Order Comparisons for Phylogenetic Inference: Evolution of the Mitochondrial Genome. In: *Proceedings of the National Academy of Sciences* 89 (1992), S. 6575–6579
- [141] SCHILIT, B. ; ADAMS, N. ; WANT, R.: Context-aware computing applications. In: *1st International Workshop on Mobile Computing Systems and Applications*, 1994
- [142] SCHILIT, Bill N. ; ADAMS, Norman ; GOLD, Rich ; TSO, Michael ; WANT, Roy: The PARCTAB mobile computing system. In: *Proceedings Fourth Workshop on Workstation Operating Systems (WWOS-IV)*, 1993
- [143] SCHILIT, Bill N. ; THEIMER, Marvin M.: Disseminating Active Map Information to Mobile Hosts. In: *IEEE Network* 8(5) (1994), September/October, S. 22–32
- [144] SCHMIDT, Albrecht: *Ubiquitous Computing – Computing in Context*, Lancaster University, Diss., November 2002

- [145] SCHMIDT, Albrecht ; AIDOO, Kofi A. ; TAKALUOMA, Antti ; TUOMELA, Urpo ; LAERHOVEN, Kristof V. ; VELDE, Walter V.: Advanced Interaction in Context. In: *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*. London, UK : Springer-Verlag, 1999. – ISBN 3–540–66550–1, S. 89–101
- [146] SCHMIDT, Albrecht ; BEIGL, Michael ; GELLERSEN, Hans-Werner: There is more to context than location. In: *Computers and Graphics Journal* 33 (1999), Nr. 6, S. 893–902
- [147] SCHMIDT, Albrecht ; GELLERSEN, Hans-Werner: Modell, Architektur und Plattform für Informationssysteme mit Kontextbezug. In: *Informatik - Forschung und Entwicklung* 16 4 (2001), November, S. 213 – 224
- [148] SCHMIDT, Albrecht ; LAERHOVEN, Kristof V.: How to build smart appliances. In: *IEEE Personal Communications* 8(4) (2001), S. 66 – 71
- [149] SCHMITT, Reinhold: Kontextualisierung und Konversationsanalyse. In: *Deutsche Sprache* 4/93 (1994), S. 326–354
- [150] SHADBOLT, Nigel ; BERNERS-LEE, Tim ; HALL, Wendy: The Semantic Web Revisited. In: *IEEE Intelligent Systems* 21(3) (2006), S. 96–101
- [151] SPEARS, William ; ANAND, V.: A Study of Crossover Operators in Genetic Programming. In: *Proceedings of the Sixth International Symposium on Methodologies for Intelligent Systems*, 1991
- [152] SRINIVASAN, R.: Distributed radar detection theory. In: *Inst. Elec. Eng*, 1986
- [153] STONE, M. H.: The Generalized Weierstrass Approximation Theorem. In: *Mathematics Magazine* 21(4)/(5) (1948), S. 167–184/237–254
- [154] STRANG, Gilbert: *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, 1986
- [155] STRANG, Thomas ; LINNHOFF-POPIEN, Claudia: A Context Modeling Survey. In: *1st International Workshop on Advanced Context Modeling, Reasoning and Management, 6th International Conference on Ubiquitous Computing*. Nottingham, UK, September 2004, S. 33–40

- [156] STRANGE, Thomas ; LINNHOFF-POPIEN, Claudia ; FRANK, Korbinian: Applications of a Context Ontology Language. In: *Proceedings of International Conference on Software, Telecommunications and Computer Networks*, 2003
- [157] STRANGE, Thomas ; LINNHOFF-POPIEN, Claudia ; FRANK, Korbinian: CoOL: A Context Ontology Language to enable Contextual Interoperability. In: *Proceedings of 4th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems*, 2003
- [158] STROHBACH, M. ; GELLERSEN, H.-W. ; KORTUEM, G. ; KRAY, Christian: Cooperative Artefacts: Assessing Real World Situations with Embedded Technology. In: *Proceedings of UBICOMP 2004*, 2004
- [159] SUGENO, M. ; KANG, G. T.: Structure identification of fuzzy model. In: *Fuzzy Sets Syst.* 28 (1988), Nr. 1, S. 15–33. – ISSN 0165–0114
- [160] SUN, J.: Mobile ad hoc networking: an essential technology for pervasive computing. In: *International Conferences on Info-tech & Info-net*, 2001
- [161] T. R. GRUBER : Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In: GUARINO, N. (Hrsg.) ; POLI, R. (Hrsg.): *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Dordrecht, The Netherlands : Kluwer Academic Publishers, 1993
- [162] T. R. GRUBER: A Translation Approach to Portable Ontology Specifications. In: *Knowledge Acquisition* 5 (2) (1993), S. 199–220
- [163] TAKAGI, T. ; SUGENO, M.: Fuzzy identification of systems and its applications to modeling and control. In: *IEEE Transactions on Systems, Man and Cybernetics* 15 (1985), S. 116–132
- [164] *AwareOffice initiative at Telecooperation Office (TecO), University of Karlsruhe*. <http://www.teco.edu/awareoffice/>. 2003
- [165] TENNEY, R. R. ; JR., N. R. S.: Detection with distributed sensors. In: *IEEE Trans. Aerospace Elect. Syst.* AES-17 (1981), S. 501–510
- [166] THOMOPOULOS, S. C. A. ; VISWANATHAN, R. ; BOUGOOU LIAS, D. C.: Optimal decision fusion in multiple sensor systems. In: *IEEE Trans. Aerospace Electron. Syst.* AES-23, (1987), S. 644–653

- [167] USUNIER, Nicolas ; AMINI, Massih-Reza ; GALLINARI, Patrick: Generalization error bounds for classifiers trained with interdependent data. In: *NIPS*, 2005
- [168] VIJAYRAGHAVAN, Vikram ; BARTON, John J.: WISE - A Simulator Toolkit for Ubiquitous Computing Scenarios. In: *Adjuncted Proceedings of UbiComp 2001*, 2001
- [169] W3C. *CC/PP Standard Ressource Page.*
<http://www.w3.org/Mobile/CCPP/>. Aug. 2006
- [170] WANG, L.-X.: Fuzzy systems are universal approximators. In: *IEEE International Conference on Fuzzy Systems*, 1992
- [171] WANG, L.-X. ; MENDEL, J. M.: Fuzzy basis function, universal approximation, and orthogonal least squares learning. In: *IEEE Trans. on Neural Networks* 3(5) (1992), S. 807–814
- [172] WANG, X.H. ; ZHANG, D.Q. ; GU, T. ; PUNG, H.K.: Ontology based context modeling and reasoning using OWL. In: *Pervasive Computing and Communications Workshops*, IEEE, March 2004, S. 18 – 22
- [173] WANT, R. ; SCHILIT, B. N. ; ADAMS, N. I. ; GOLD, R. ; PETERSEN, K. ; GOLDBERG, D. ; ELLIS, J. R. ; WEISER, M.: An overview of the PARCTAB ubiquitous computing experiment. In: *IEEE Personal Communications* 2 (1995), Dec, Nr. 6, S. 28–33
- [174] WARD, Andy ; JONES, Alan ; HOPPER, Andy: A new location technique for the active office. In: *IEEE Personal Communications* 4(5) (1997), October, S. 42–47
- [175] WEIERSTRASS, Karl: Über die analytische Darstellbarkeit sogenannter willkürlicher Funktionen einer reellen Veränderlichen. In: *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin* (II) (1885), S. 789–805
- [176] WEISER, M.: The computer for the 21st century. In: *Scientific American* 265 (3) (1991), S. 66–75
- [177] WEISER, Mark ; BROWN, John S.: Designing Calm Technology / Xerox PARC. 1996. – Forschungsbericht

- [178] YAGER, R.R. ; FILEV, D.P.: Generation of fuzzy rules by mountain clustering. In: *Intelligent and Fuzzy Systems* 2 (1994), S. 209–219
- [179] YAU, S. S. ; KARIM, F. ; WANG, Y. ; WANG, B. ; S.GUPTA: Reconfigurable Context-Sensitive Middleware for Pervasive Computing. In: *IEEE Pervasive Computing* 1(3) (2002), S. 33–40
- [180] YAU, S.S. ; HUANG, D. ; GONG, H. ; SETH, S: Development and runtime support for situation-aware application software in ubiquitous computing environments. In: *Computer Software and Applications Conference*, 2004, S. 452 – 457
- [181] YAU, S.S. ; WANG, Yu ; KARIM, F.: Development of situation-aware application software for ubiquitous computing environments. In: *Computer Software and Applications Conference*, 2002, S. 233 – 238
- [182] ZADEH, L. A.: Outline of a new approach to the analysis of complex systems and descision processes. In: *IEEE Trans. on Systems, Man and Cybernetics* 3(1) (1973), S. 28–44
- [183] ZELL, Andreas ; MAMIER, Günter ; VOGT, Michael ; MACHE, Niels ; HÜBNER, Ralf ; DÖRING, Sven ; HERRMANN, Kai-Uwe ; SOYEZ, Tobias ; SCHMALZL, Michael ; SOMMER, Tilman ; HATZIGEORGIOU, Artemis ; POSSELT, Dietmar ; SCHREINER, Tobias ; KETT, Bernward ; CLEMENTE, Gianfranco ; WIELAND, Jens: *SNNS: Stuttgart Neural Network Simulator User Manual, Version 4.1*. Universität Stuttgart
- [184] ZIMMER, Tobias: *Diversified Communication in Ubiquitous Computing*. Germany, University of Karlsruhe, Diplomarbeit, 2001
- [185] ZIMMER, Tobias: Towards a Better Understanding of Context Attributes. In: *Proceedings of PerCom 2004*. Orlando, USA, März 2004, S. 23–28
- [186] ZIMMER, Tobias. *MemoClip II Project*. <http://www.teco.edu/~zimmer/memoclip.php>. 2006
- [187] ZIMMER, Tobias: QoC: Quality of Context - Improving the Performance of Context-Aware Applications. In: *Adjuned Proceedings of Pervasive 2006*, 2006
- [188] ZIMMER, Tobias: Quality of Context: Handling Context Dependancies. In: *Adjuned Proceedings of Ubicomp 2006*, 2006

- [189] ZIMMER, Tobias: Kostenmodell der Kontextkommunikation und Kontextverarbeitung in Ubiquitären Informationsumgebungen / University of Karlsruhe. 2007 (2007/02). – ISSN 1432-7864
- [190] ZIMMER, Tobias ; BEIGL, Michael: AwareOffice: Integrating Modular Context-Aware Applications. In: *Proceedings of the 6th International Workshop on Smart Appliances and Wearable Computing (IWSAWC)*, IEEE Computer Society Press, 2006
- [191] ZIMMERMANN, Hubert: OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection. In: *IEEE Transactions on Communications* 28(4) (1980), S. 425–432

Abbildungsverzeichnis

1.1	Reduzierter Verarbeitungsgraph einer ubiquitären Informationsumgebung	5
1.2	Vollständiger Verarbeitungsgraph	5
1.3	Aufbau dieser Arbeit	9
2.1	Einfache Architektur heutiger kontextsensitiver Anwendungen . .	21
3.1	Übergang vom monolithischen zum Schichtenmodell	39
3.2	Kontextmanagement als Schnittstelle zwischen Kommunikation und Kontextverarbeitung	47
3.3	Produzent-Konsument-Schema: Artefaktsicht	55
3.4	Produzent-Konsument-Schema: Kontextsicht	55
3.5	Hierarchische Anordnung von Kontexten im dynamischen Verarbeitungsmodell	58
3.6	Temporale Entwicklung von Kontexten	59
3.7	Entkoppelte Kommunikation: CSpace	60
3.8	Alter und Ableitungsdauer von Kontexten	68
3.9	Zeitlicher Wechsel: Kontext A \rightarrow Kontext B	74
3.10	Idealer Erkenner: Wechsel Kontext A \rightarrow Kontext B	75
3.11	Verlaufsmodell der Verlässlichkeit von Kontext	75
3.12	Auffächerung und Multiplikation	78
3.13	Zyklische Verarbeitung	81
4.1	Architektur: Artefakt	87
4.2	Architektur: AwareCon	88
4.3	ConCom: Kontexttupel bilden ConCom-Sätze	89
4.4	Architektur: Kontextmanagementschicht	92
4.5	Kontextattribute im Feld A eines Kontextes C	95
4.6	Attribut Kontextalter (CA) mit Zeitstempeln: Context Seed Age (CSA) und Context Production Age (CPA)	96

4.7	Context Seed Age (CSA) und Context Production Age (CPA) eines Kontextes C	97
4.8	Kontextalter verschiedener Kontexte eines Ableitungsbaumes . . .	98
4.9	Attribut räumliche Herkunft (LOC)	100
4.10	Beispiel eines Lokationsbaumes	102
4.11	Aufbau der RAUM-Lokationsbeschreibung als Kontextattribut . .	103
4.12	Attribut Verlässlichkeit von Kontexten (FCR)	104
4.13	Verlauf der Verlässlichkeit von Kontext	105
4.14	Verlässlichkeit von Kontextclustern	106
4.15	Schematische Architektur des FCR-Systems	107
4.16	4-Schicht-ANFIS-Architektur [20]	109
4.17	Attribut Abhängigkeit von Kontexten (GRC)	118
4.18	links: one point Crossover; rechts: two point Crossover	122
4.19	Cut and Slice Crossover	123
4.20	Vererbung von zwei Elterngenomen	124
4.21	Vererbung von drei Elterngenomen	125
4.22	Stufenweise Vererbung	127
4.23	Mehrfachvererbung	127
4.24	Stufenweise Vererbung über 10 Generationen	128
4.25	Mehrfachvererbung über 10 Generationen	129
4.26	Dubletten-Fehler können durch Vererbung verschwinden	130
4.27	$P(terr)$ mit 256 verschiedenen Allelen (75 Eltern)	132
4.28	$P(terr)$ mit 256 verschiedenen Allelen (10 Eltern)	133
4.29	$E(rel_{err})$ mit 256 verschiedenen Allelen, bis 100 Eltern	135
5.1	AwarePens: Tafelstifte mit pParts	152
5.2	AwareCon mit Kontextmanagementschicht	153
5.3	Erweiterter ACL-Protokollkopf	154
5.4	AwareSponge: kontextsensitiver Tafelschwamm	155
5.5	Stuhl mit uPart	156
5.6	Projektor mit uPart	156
5.7	Fenster mit uPart	157
5.8	Tür mit uPart	157
5.9	Interaktives Türschild am AwareOffice	158
6.1	Verteilung der Verwandtschaftsgrade für $n = 10, 30, 50$	163
6.2	Verteilung der Verwandtschaftsgrade für $n = 100, n = 500$ und $n = 1000$	164

6.3	Erwarteter Fehler in Abhängigkeit von der Anzahl der Allele . . .	166
6.4	Simulation CPA: Subskriptionsgraph	168
6.5	Simulation CPA: Ableitungsgraph	168
6.6	Erkennungsrate CPA: aktiviert (1 Schritt) und deaktiviert	169
6.7	Erkennungsraten CSA: aktiviert (4 Schritte) und deaktiviert . . .	172
6.8	Erkennungsraten CSA: aktiviert (2, 3 und 4 Schritte)	174
6.9	Simulation LOC: Subskriptionsgraph	175
6.10	Simulation LOC: Ableitungsgraph	175
6.11	Erkennungsrate LOC: aktiviert und deaktiviert	177
6.12	Simulation FCR: Subskriptions- und Ableitungsgraph	178
6.13	Erkennungsrate FCR: aktiviert und deaktiviert	179
6.14	Simulation GRC: Auffächerung und Multiplikation	182
6.15	Erkennungsrate GRC: aktiviert und deaktiviert	183
6.16	Simulation GRC: zyklische Vererbung	185
6.17	Erkennungsrate GRC: aktiviert und deaktiviert	186
6.18	Erkennungsraten: links ohne, rechts mit Kontextmanagement . . .	191
6.19	Mittlere Erkennungsraten: ohne und mit Kontextmanagement . . .	192
A.1	Simulation Gesamtszenario: Subskriptionsgraph	263

Tabellenverzeichnis

2.1	Übersicht der Eigenschaften der Anwendungsdomäne des Kontextmanagementsystems	17
3.1	Kosten im Vergleich [189]	42
6.1	Auflösungsvermögen von GRC	165
6.2	Feldversuch FCR: Ergebnisse	197

Glossar

ubiquitäre Informationsumgebung: Eine ubiquitäre Informationsumgebung ist ein nicht zwingend zusammenhängender Bereich oder ein Gebiet physikalischer Ausdehnung, in dem Artefakte und optional kontextsensitive Infrastrukturdienste vorhanden sind und miteinander kommunizieren können, Seite 47.

ubiquitäre Informationsumgebung: Eine ubiquitäre Informationsumgebung ist ein räumlicher Bereich, in dem Ubiquitous Computing Systeme arbeiten und entsprechende Dienste bereitstellen, Seite 15.

A/D-Wandler: siehe Analog-Digital-Wandler.

Ableitungsbaum: Unter dem Ableitungsbaum eines Kontextes versteht man die Baumstruktur, welche alle Kontexte enthält, die zu einem Zielkontext beitragen., Seite 96.

Aggregation: Die Aggregation von Daten bezeichnet die algorithmische Verknüpfung von verschiedenartigen Daten zum Zweck der Gewinnung neuer Erkenntnisse, Seite 13.

Analog-Digital-Wandler: Ein Analog-Digital-Wandler (A/D-Wandler) ist eine elektronische Schaltung, mit deren Hilfe ein analoger Signalverlauf sowohl in der Zeit als auch in der Amplitude quantisiert werden kann. Die Ausgabe eines A/D-Wandlers ist ein digitaler Datenstrom, der das ursprüngliche analoge Signal repräsentiert, Seite 48.

Artefakt: Ein Artefakt ist ein computerisierter Alltagsgegenstand in einer ubiquitären Informationsumgebung, der eine kontextsensitive Anwendung beinhaltet. Der Begriff wird oft synonym für die Anwendung und ihre Funktionalität verwendet, wenn die Gefahr einer Verwechslung nicht besteht, Seite 47.

Attribut Fuzzy Context Reliability: Fuzzy Context Reliability (FCR) ist ein Kontextattribut, welches die Bestimmung der semantischen Verlässlichkeit von Kontexten in einer ubiquitären Informationsumgebung erlaubt, Seite 104.

Attribut Genetic Relation of Context: Genetic Relation of Context (GRC) ist ein Kontextattribut, welches die quantitative Bestimmung des Verwandtschaftsgrades von Kontexten in einer ubiquitären Informationsumgebung erlaubt, Seite 118.

Attribut Kontextalter: Das Kontextalter (CA) ist ein Kontextattribut, welches die Bestimmung temporaler Parameter von Kontexten in einer ubiquitären Informationsumgebung erlaubt, Seite 96.

Attribut räumliche Herkunft: Die räumliche Herkunft (LOC) ist ein Kontextattribut, welches die Bestimmung der räumlichen Herkunft von Kontexten in einer ubiquitären Informationsumgebung erlaubt, Seite 100.

Auffächerung und Multiplikation: Auffächerung und Multiplikation stellt ein Grundmuster der informationellen Abhängigkeit von Kontexten in einer ubiquitären Informationsumgebung dar, Seite 77.

CA: siehe Attribut Kontextalter.

Computer Supported Cooperative Work: Computerunterstütztes kooperatives Arbeiten (engl.: Computer Supported Cooperative Work (CSCW)) beschäftigt sich als Forschungsbereich des Ubiquitous Computing mit Techniken zur Kollaborationsunterstützung im Arbeitsumfeld, Seite 15.

Computerunterstütztes kooperatives Arbeiten: siehe Computer Supported Cooperative Work.

Context Production Age: (CPA) Ein Zeitstempel der angibt, wann ein Kontext produziert wurde, Seite 96.

Context Seed Age: (CSA) Ein Zeitstempel, der das Alter des ältesten in einem Kontext enthaltenen Datums angibt, Seite 96.

CPA: siehe Context Production Age.

CSA: siehe Context Seed Age.

CSCW: siehe Computer Supported Cooperative Work.

FCR: siehe Attribut Fuzzy Context Reliability.

Fusion: Die Fusion von Daten bezeichnet die algorithmische Verknüpfung gleichartiger Daten zum Zweck der Genauigkeitsverbesserung oder Fehlerreduktion, Seite 13.

GRC: siehe Attribut Genetic Relation of Context.

kognitiver Kontext: Kontext, wie er vom Menschen wahrgenommen wird. Der kognitive Kontext beinhaltet sowohl messbare physikalische Umweltfaktoren als auch nicht messbare Faktoren, wie die Erfahrungen oder die Intention des Menschen, Seite 49.

Kontext: Kontext bezeichnet die Einheit aus Kontextdaten und Kontextattributen. Des Weiteren besitzt Kontext einen Typ, der durch die enthaltenen Kontextdaten festgelegt wird (siehe auch *semantischer Kontext* und *kognitiver Kontext*), Seite 51.

Kontextaggregation: siehe Aggregation.

Kontextalgorithmen: Oberbegriff für Algorithmen, die in der Kontexterkenennung, Kontextaggregation und Kontextfusion Verwendung finden, Seite 14.

Kontextattribut: Ein Kontextattribut ist eine Eigenschaft eines Kontextdatums, die von dessen semantischer Bedeutung unabhängig ist, Seite 51.

Kontextdaten: Kontextdaten setzen sich aus mindestens einem Sensordatum und einer Metainformation zusammen. Kontextdaten haben eine semantische Bedeutung bezüglich der aktuellen Situation, Seite 50.

Kontexterkennungsrate: Die Kontexterkennungsrate gibt das durchschnittliche Verhältnis zwischen korrekt erkannten Kontexten und der Anzahl der real vorhandenen, verschiedenen Kontexte an, Seite 72.

Kontextfusion: siehe Fusion.

Kontextinstanz: Eine Kontextinstanz ist Kontext einer bestimmten Kontextklasse, dessen Attributen konkrete Werte zugeordnet sind und dessen Kontextdaten einen konkreten Wert besitzen, Seite 52.

Kontextklasse: Eine Kontextklasse wird durch den Typ des Kontextes induziert. Die Kontextklasse hat alle Eigenschaften und Attribute eines Kontextes, ohne dass diesen konkrete Werte zugewiesen sind, Seite 52.

Kontextkonsument: Ein Artefakt, welches Kontexte aus einer ubiquitären Informationsumgebung aufnimmt und verarbeitet, heißt Konsument dieser Kontexte, Seite 54.

Kontextproduzent: Ein Artefakt, welches einen Kontext erzeugt und diesen in einer ubiquitären Informationsumgebung zur Verfügung stellt, heißt Produzent dieses Kontextes, Seite 54.

Kontextqualität: Kontextqualität sind Informationen, auf welchen Metriken definiert werden können, die eine quantitative Beurteilung der Qualität von Kontexten bezüglich bestimmter Qualitätsparameter zulassen. Kontextqualität bezieht sich auf einzelne Kontexte, nicht auf die Prozesse, durch die Kontexte bereitgestellt werden. Kontextqualität ist ein globales Maß, das auf alle Kontexte gleichermaßen anwendbar ist; ihrer Bemessung werden für alle Kontexte die gleichen Qualitätsparameter zu Grunde gelegt, Seite 36.

Kontextsensitivität: Ein System oder eine Anwendung ist kontextsensitiv, wenn es Kontextinformationen verarbeitet und sein/ihr inneres und/oder äußeres Verhalten aufgrund der verarbeiteten Kontextinformation ändern kann, Seite 20.

LOC: siehe Attribut räumliche Herkunft.

Metainformation: Metainformation bezeichnet solche Daten, die nicht innerhalb des Informationssystems selbst gewonnen werden können, sondern stattdessen von außen, zum Beispiel durch den Entwickler, bereitgestellt werden müssen, Seite 48.

PMSC: siehe Probabilistic Multi-Site Crossover.

Probabilistic Multi-Site Crossover: Probabilistic Multi-Site Crossover (PMSC) ist Teil des GRC-Algorithmensystems. PMSC stellt eine spezielle Vererbungs-technik für Kontextgenome dar, Seite 121.

semantischer Kontext: Die Repräsentation des kognitiven Kontextes in einem kontextsensitiven System heißt semantischer Kontext. Der semantische

Kontext dient der Erfassung der Bedeutung einer Kontextinformation. Semantische Kontextmodelle modellieren diese Bedeutung und die Beziehung zwischen semantischen Kontexten, Seite 50.

Sensordaten: Sensordaten sind Daten, die ein Artefakt oder ein Infrastrukturdienst durch physikalische Umweltsensoren erfassen kann, Seite 48.

TCDT: siehe Total Context Derive Time.

Total Context Derive Time: (*TCDT*) Zeit zum Ableiten eines Kontextes von der Erhebung des ersten Datums bis zu seiner Erzeugung, Seite 99.

Ubicomp: Kurzform von Ubiquitous Computing. Forschungsbereich der modernen Informatik, Seite 2.

Verarbeitungsgraph: Graph, der gemäß des Produzent-Konsument-Schemas (vgl. Abschnitt 3.5) die Interaktion von Artefakten durch den Austausch von Kontexten repräsentiert, Seite 55.

Verlässlichkeit von Kontext: Die Verlässlichkeit von Kontext stellt ein Maß für die inhaltliche Qualität einer Kontextinstanz dar. Sie ist somit ein Kontextattribut, da sie jedem Kontext, unabhängig von seiner semantischen Bedeutung, ein Maß seiner inhaltlichen Qualität zuordnet., Seite 73.

Verwandtschaft von Kontexten: Die Verwandtschaft von Kontexten ist ein Maß für die informationelle Abhängigkeit von Kontextinformationen, die durch den Prozess der Ableitung von Kontexten höherer Ordnung entsteht, Seite 77.

zyklische Verarbeitung: Zyklische Verarbeitung ist ein Grundmuster der informationellen Abhängigkeit von Kontexten in einer ubiquitären Informationsumgebung, Seite 80.

Anhang A

Simulationsergebnisse

Dieser Anhang enthält die ausführlichen Simulationsergebnisse aus der Evaluierung der Kontextattribute mit context_sim. Jeder der folgenden Abschnitte enthält die Auswertung eines Simulationslaufes. Dabei ist hier für jeden, der Einzelversuche in Abschnitt 6.2 ein Simulationslauf aufgezeichnet.

Die Daten enthalten Statistiken für alle an der Simulation beteiligten Artefakte, sowie zusammenfassende Resultate des Simulationslaufes. Einige der hier vollständig wiedergegebenen Statistiken wurden bereits im Kapitel Evaluierung im Detail besprochen.

Der Abschnitt A.5 enthält zusätzlich zusammenfassende Statistiken zu dem Versuch „Gesamtszenario“ und dessen Simulationsgraph.

A.1 Simulation: CA

A.1.1 CPA ohne Filter

```
-----  
List of Artefacts  
-----
```

```
1 Artefact Typ_A_0 of type 1  
2 Artefact Typ_A_1 of type 1  
3 Artefact Typ_A_2 of type 1  
4 Artefact Typ_A_3 of type 1  
5 Artefact Typ_A_4 of type 1  
6 Artefact Typ_A_5 of type 1  
7 Artefact Typ_A_6 of type 1  
8 Artefact Typ_A_7 of type 1  
9 Artefact Typ_A_8 of type 1  
10 Artefact Typ_A_9 of type 1  
11 Artefact Typ_B_0 of type 2  
-----  
-----
```

Anhang A Simulationsergebnisse

Stats for Artefact Typ_A_0:

Artefact type: 1

Total active steps: 1221
Total number of contexts consumed: 0
Total number of contexts produced: 1221
Overall recognition rate: 0.8688279075578874
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A_1:

Artefact type: 1

Total active steps: 1244
Total number of contexts consumed: 0
Total number of contexts produced: 1244
Overall recognition rate: 0.8720241144770712
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A_2:

Artefact type: 1

Total active steps: 1315
Total number of contexts consumed: 0
Total number of contexts produced: 1315
Overall recognition rate: 0.8722688720057123
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A_3:

Artefact type: 1

Total active steps: 1243
Total number of contexts consumed: 0
Total number of contexts produced: 1243
Overall recognition rate: 0.8699538165348049
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A_4:

Artefact type: 1

Total active steps: 1212
Total number of contexts consumed: 0
Total number of contexts produced: 1212
Overall recognition rate: 0.868199061386923
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A_5:

Artefact type: 1

Total active steps: 1255
Total number of contexts consumed: 0
Total number of contexts produced: 1255
Overall recognition rate: 0.8744031260687536
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A_6:

Artefact type: 1

Total active steps: 1261
Total number of contexts consumed: 0
Total number of contexts produced: 1261
Overall recognition rate: 0.8699661757180597
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A_7:

Artefact type: 1

Total active steps: 1254
Total number of contexts consumed: 0
Total number of contexts produced: 1254
Overall recognition rate: 0.8721753968686289
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A_8:

Artefact type: 1

Total active steps: 1240
Total number of contexts consumed: 0
Total number of contexts produced: 1240
Overall recognition rate: 0.8704691086900058

Anhang A Simulationsergebnisse

Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A_9:

Artefact type: 1

Total active steps: 1278
Total number of contexts consumed: 0
Total number of contexts produced: 1278
Overall recognition rate: 0.869083255354262
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_B_0:

Artefact type: 2

Total active steps: 5000
Total number of contexts consumed: 14994
Total number of contexts produced: 4998
Overall recognition rate: 0.3367674651174112
Mean relatedness of consumed Cs: 0.003897114401316082
Var of relatedness of consumed Cs: 4.254780673105989E-5

General Context Stats

Total number of contexts consumed: 14994
Total number of contexts produced: 17521
Mean derive time: 4.227490996398555
Max internal context reliability: 0.9999296292129205
Min internal context reliability: 0.10441586267545028
Mean internal context reliability: 0.7184307631313341

Simulation stats

Total number of simulation steps: 5000
Total number of simTicks: 5000
Total number of artefacts: 11
Total number of contexts in CSpace: 21
Total number of derivations: 17521.0
Mean number of derivations per artefact: 1592.8181818181818
Max recognition rate: 0.8744031260687536
Min recognition rate: 0.3367674651174112
Mean recognition rate: 0.822
Var of recognition rate: 0.025923684300174187
Min Seed Reliability: 0.6841621983099551
Max Seed Reliability: 0.9999296292129205

A.1.2 CPA mit Filter

List of Artefacts

1 Artefact Typ_A_0 of type 1
2 Artefact Typ_A_1 of type 1
3 Artefact Typ_A_2 of type 1
4 Artefact Typ_A_3 of type 1
5 Artefact Typ_A_4 of type 1
6 Artefact Typ_A_5 of type 1
7 Artefact Typ_A_6 of type 1
8 Artefact Typ_A_7 of type 1
9 Artefact Typ_A_8 of type 1
10 Artefact Typ_A_9 of type 1
11 Artefact Typ_B_0 of type 2

Stats for Artefact Typ_A_0:

Artefact type: 1

Total active steps: 1267
Total number of contexts consumed: 0
Total number of contexts produced: 1267
Overall recognition rate: 0.8694542335800342
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A_1:

Artefact type: 1

Total active steps: 1260
Total number of contexts consumed: 0
Total number of contexts produced: 1260
Overall recognition rate: 0.8665496374519459
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A_2:

Artefact type: 1

Anhang A Simulationsergebnisse

Total active steps: 1248
Total number of contexts consumed: 0
Total number of contexts produced: 1248
Overall recognition rate: 0.8672557146655271
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A_3:

Artefact type: 1

Total active steps: 1242
Total number of contexts consumed: 0
Total number of contexts produced: 1242
Overall recognition rate: 0.8688679894229405
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A_4:

Artefact type: 1

Total active steps: 1298
Total number of contexts consumed: 0
Total number of contexts produced: 1298
Overall recognition rate: 0.8712696510558379
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A_5:

Artefact type: 1

Total active steps: 1246
Total number of contexts consumed: 0
Total number of contexts produced: 1246
Overall recognition rate: 0.8703291534651894
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A_6:

Artefact type: 1

Total active steps: 1224
Total number of contexts consumed: 0
Total number of contexts produced: 1224
Overall recognition rate: 0.8687347343658571

Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A_7:

Artefact type: 1

Total active steps: 1207
Total number of contexts consumed: 0
Total number of contexts produced: 1207
Overall recognition rate: 0.8692057307181043
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A_8:

Artefact type: 1

Total active steps: 1210
Total number of contexts consumed: 0
Total number of contexts produced: 1210
Overall recognition rate: 0.8687956732887065
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A_9:

Artefact type: 1

Total active steps: 1311
Total number of contexts consumed: 0
Total number of contexts produced: 1311
Overall recognition rate: 0.8688477738381968
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_B_0:

Artefact type: 2

Total active steps: 5000
Total number of contexts consumed: 7203
Total number of contexts produced: 2401
Overall recognition rate: 0.6562353088040962
Mean relatedness of consumed Cs: 0.0038178536720810848
Var of relatedness of consumed Cs: 4.112779654567796E-5

General Context Stats

Total number of contexts consumed: 7203
Total number of contexts produced: 14914
Mean derive time: 1.0
Max internal context reliability: 0.9995932254875298
Min internal context reliability: 0.4760878449588716
Mean internal context reliability: 0.834695197039677

Simulation stats

Total number of simulation steps 5000
Total number of simTicks: 5000
Total number of artefacts: 11
Total number of contexts in CSpace: 10
Total number of derivations: 14914.0
Mean number of derivations per artefact: 1355.8181818181818
Max recognition rate: 0.8712696510558379
Min recognition rate: 0.6562353088040962
Mean recognition rate: 0.85
Var of recognition rate: 0.004114310899705806
Min Seed Reliability: 0.6782086681406494
Max Seed Reliability: 0.9995932254875298

A.1.3 CSA ohne Filter

List of Artefacts

1 Artefact Typ_A_0 of type 1
2 Artefact Typ_B_0 of type 2
3 Artefact Typ_C_0 of type 3
4 Artefact Typ_D_0 of type 4

Stats for Artefact Typ_A_0:

Artefact type: 1

Total active steps: 1262
Total number of contexts consumed: 0
Total number of contexts produced: 1262
Overall recognition rate: 0.8666473273826618
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_B_0:

Artefact type: 2

Total active steps: 1238
Total number of contexts consumed: 1076
Total number of contexts produced: 1076
Overall recognition rate: 0.5978200883738743
Mean relatedness of consumed Cs: 0.29033457249070616
Var of relatedness of consumed Cs: 0.20408220320259748

Stats for Artefact Typ_C_0:

Artefact type: 3

Total active steps: 1271
Total number of contexts consumed: 1001
Total number of contexts produced: 1001
Overall recognition rate: 0.42404322114840154
Mean relatedness of consumed Cs: 0.417382617382617
Var of relatedness of consumed Cs: 0.24167976467976504

Stats for Artefact Typ_D_0:

Artefact type: 4

Total active steps: 1277
Total number of contexts consumed: 932
Total number of contexts produced: 932
Overall recognition rate: 0.3011838563535718
Mean relatedness of consumed Cs: 0.4768240343347637
Var of relatedness of consumed Cs: 0.2482867397391905

General Context Stats

Total number of contexts consumed: 3009
Total number of contexts produced: 4271
Mean derive time: 7.854436689930223
Max internal context reliability: 0.999
Min internal context reliability: 0.06118118473419158
Mean internal context reliability: 0.5717944183419051

Simulation stats

Total number of simulation steps 5000
Total number of simTicks: 5000
Total number of artefacts: 4

```
Total number of contexts in CSpace:      5
Total number of derivations:             4271.0
Mean number of derivations per artefact:  1067.75
Max recognition rate:                     0.8666473273826618
Min recognition rate:                     0.3011838563535718
Mean recognition rate:                     0.547
Var of recognition rate:                   0.06010010780037284
Min Seed Reliability:                     0.6970650142023073
Max Seed Reliability:                     0.999
-----
```

A.1.4 CSA mit Filter 444

```
-----
List of Artefacts
-----
```

```
1 Artefact Typ_A_0 of type 1
2 Artefact Typ_B_0 of type 2
3 Artefact Typ_C_0 of type 3
4 Artefact Typ_D_0 of type 4
-----
```

```
-----
Stats for Artefact Typ_A_0:
-----
```

```
Artefact type: 1
```

```
Total active steps:           1265
Total number of contexts consumed:  0
Total number of contexts produced: 1265
Overall recognition rate:         0.8689419133045181
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN
-----
```

```
-----
Stats for Artefact Typ_B_0:
-----
```

```
Artefact type: 2
```

```
Total active steps:           1235
Total number of contexts consumed: 868
Total number of contexts produced: 868
Overall recognition rate:         0.7397653608461507
Mean relatedness of consumed Cs: 0.23221966205837163
Var of relatedness of consumed Cs: 0.17565043073944733
-----
```

```
-----
Stats for Artefact Typ_C_0:
-----
```

```
Artefact type: 3
```

Total active steps: 1249
Total number of contexts consumed: 265
Total number of contexts produced: 265
Overall recognition rate: 0.7614448044067432
Mean relatedness of consumed Cs: 0.18075471698113194
Var of relatedness of consumed Cs: 0.14535633060161354

Stats for Artefact Typ_D_0:

Artefact type: 4

Total active steps: 1236
Total number of contexts consumed: 51
Total number of contexts produced: 51
Overall recognition rate: 0.8174269582507337
Mean relatedness of consumed Cs: 0.08104575163398695
Var of relatedness of consumed Cs: 0.07338910675381263

General Context Stats

Total number of contexts consumed: 1184
Total number of contexts produced: 2449
Mean derive time: 2.700168918918917
Max internal context reliability: 0.999
Min internal context reliability: 0.5218457917658359
Mean internal context reliability: 0.810453042704471

Simulation stats

Total number of simulation steps: 5000
Total number of simTicks: 5000
Total number of artefacts: 4
Total number of contexts in CSpace: 4
Total number of derivations: 2449.0
Mean number of derivations per artefact: 612.25
Max recognition rate: 0.8689419133045181
Min recognition rate: 0.7397653608461507
Mean recognition rate: 0.797
Var of recognition rate: 0.003377610354511957
Min Seed Reliability: 0.6858631562865608
Max Seed Reliability: 0.999

A.1.5 CSA mit Filter 234

List of Artefacts

1 Artefact Typ_A_0 of type 1
2 Artefact Typ_B_0 of type 2
3 Artefact Typ_C_0 of type 3
4 Artefact Typ_D_0 of type 4

Stats for Artefact Typ_A_0:

Artefact type: 1

Total active steps: 1213
Total number of contexts consumed: 0
Total number of contexts produced: 1213
Overall recognition rate: 0.8716481617599818
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_B_0:

Artefact type: 2

Total active steps: 1252
Total number of contexts consumed: 566
Total number of contexts produced: 566
Overall recognition rate: 0.8257069041902717
Mean relatedness of consumed Cs: 0.1059481743227327
Var of relatedness of consumed Cs: 0.09154961068201012

Stats for Artefact Typ_C_0:

Artefact type: 3

Total active steps: 1272
Total number of contexts consumed: 207
Total number of contexts produced: 207
Overall recognition rate: 0.8119807498050482
Mean relatedness of consumed Cs: 0.10595813204508865
Var of relatedness of consumed Cs: 0.0908435084866772

Stats for Artefact Typ_D_0:

Artefact type: 4

Total active steps: 1221

```
Total number of contexts consumed: 71
Total number of contexts produced: 71
Overall recognition rate: 0.8050230458877926
Mean relatedness of consumed Cs: 0.11877934272300478
Var of relatedness of consumed Cs: 0.10021372680527611
```

General Context Stats

```
Total number of contexts consumed: 844
Total number of contexts produced: 2057
Mean derive time: 1.9798578199052126
Max internal context reliability: 0.999
Min internal context reliability: 0.6748421711869996
Mean internal context reliability: 0.8507029554954924
```

Simulation stats

```
Total number of simulation steps 5000
Total number of simTicks: 5000
Total number of artefacts: 4
Total number of contexts in CSpace: 6
Total number of derivations: 2057.0
Mean number of derivations per artefact: 514.25
Max recognition rate: 0.8716481617599818
Min recognition rate: 0.8050230458877926
Mean recognition rate: 0.829
Var of recognition rate: 8.978620178127574E-4
Min Seed Reliability: 0.6884374790385224
Max Seed Reliability: 0.999
```

A.2 Simulation: LOC

A.2.1 LOC ohne Filter

List of Artefacts

```
1 Artefact Typ_A1_0 of type 1
2 Artefact Typ_A1_1 of type 1
3 Artefact Typ_A2_0 of type 2
4 Artefact Typ_A2_1 of type 2
5 Artefact Typ_B_0 of type 3
```

Anhang A Simulationsergebnisse

Stats for Artefact Typ_A1_0:

Artefact type: 1

Total active steps: 4461
Total number of contexts consumed: 0
Total number of contexts produced: 4461
Overall recognition rate: 0.8692876863186606
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A1_1:

Artefact type: 1

Total active steps: 4507
Total number of contexts consumed: 0
Total number of contexts produced: 4507
Overall recognition rate: 0.8696614401999461
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A2_0:

Artefact type: 2

Total active steps: 4483
Total number of contexts consumed: 0
Total number of contexts produced: 4483
Overall recognition rate: 0.10064024159679986
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A2_1:

Artefact type: 2

Total active steps: 4508
Total number of contexts consumed: 0
Total number of contexts produced: 4508
Overall recognition rate: 0.10120858371293225
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_B_0:

Artefact type: 3


```
Total active steps: 5000
Total number of contexts consumed: 9950
Total number of contexts produced: 4975
Overall recognition rate: 0.18149978575556255
Mean relatedness of consumed Cs: 0.0037051926298157455
Var of relatedness of consumed Cs: 1.236498356501267E-4
```

General Context Stats

```
Total number of contexts consumed: 9950
Total number of contexts produced: 22934
Mean derive time: 1.0
Max internal context reliability: 0.9991578270705341
Min internal context reliability: 1.0E-6
Mean internal context reliability: 0.418934176865746
```

Simulation stats

```
Total number of simulation steps 5000
Total number of simTicks: 5000
Total number of artefacts: 5
Total number of contexts in CSpace: 5
Total number of derivations: 22934.0
Mean number of derivations per artefact: 4586.8
Max recognition rate: 0.8696614401999461
Min recognition rate: 0.10064024159679986
Mean recognition rate: 0.424
Var of recognition rate: 0.16611409316007503
Min Seed Reliability: 1.084700059263699E-4
Max Seed Reliability: 0.9991578270705341
```

A.2.2 LOC mit Filter

List of Artefacts

```
1 Artefact Typ_A1_0 of type 1
2 Artefact Typ_A1_1 of type 1
3 Artefact Typ_A2_0 of type 2
4 Artefact Typ_A2_1 of type 2
5 Artefact Typ_B_0 of type 3
```

Stats for Artefact Typ_A1_0:

Artefact type: 1

Total active steps: 4500
Total number of contexts consumed: 0
Total number of contexts produced: 4500
Overall recognition rate: 0.8698704692827414
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A1_1:

Artefact type: 1

Total active steps: 4477
Total number of contexts consumed: 0
Total number of contexts produced: 4477
Overall recognition rate: 0.8690120248366154
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A2_0:

Artefact type: 2

Total active steps: 4495
Total number of contexts consumed: 0
Total number of contexts produced: 4495
Overall recognition rate: 0.09998766892767968
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_A2_1:

Artefact type: 2

Total active steps: 4526
Total number of contexts consumed: 0
Total number of contexts produced: 4526
Overall recognition rate: 0.10068747950287961
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_B_0:

Artefact type: 3

Total active steps: 5000
Total number of contexts consumed: 8068
Total number of contexts produced: 4034

Overall recognition rate: 0.7562789611891461
Mean relatedness of consumed Cs: 0.003858866303090404
Var of relatedness of consumed Cs: 1.297455115005935E-4

General Context Stats

Total number of contexts consumed: 8068
Total number of contexts produced: 22032
Mean derive time: 1.0
Max internal context reliability: 0.9993856050618406
Min internal context reliability: 9.982021575231925E-5
Mean internal context reliability: 0.5338130619309521

Simulation stats

Total number of simulation steps 5000
Total number of simTicks: 5000
Total number of artefacts: 5
Total number of contexts in CSpace: 5
Total number of derivations: 22032.0
Mean number of derivations per artefact: 4406.4
Max recognition rate: 0.8698704692827414
Min recognition rate: 0.09998766892767968
Mean recognition rate: 0.539
Var of recognition rate: 0.16261072585812666
Min Seed Reliability: 9.982021575231925E-5
Max Seed Reliability: 0.9993856050618406

A.3 Simulation: FCR

A.3.1 FCR ohne Filter

List of Artefacts

1 Artefact Typ_A_0 of type 1
2 Artefact Typ_B_0 of type 2
3 Artefact Typ_C_0 of type 3
4 Artefact Typ_D_0 of type 4
5 Artefact Typ_E_0 of type 5

Stats for Artefact Typ_A_0:

Anhang A Simulationsergebnisse

Artefact type: 1

Total active steps: 5000
Total number of contexts consumed: 0
Total number of contexts produced: 5000
Overall recognition rate: 0.8712649900538805
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_B_0:

Artefact type: 2

Total active steps: 5000
Total number of contexts consumed: 0
Total number of contexts produced: 5000
Overall recognition rate: 0.869029681450718
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_C_0:

Artefact type: 3

Total active steps: 5000
Total number of contexts consumed: 9998
Total number of contexts produced: 4999
Overall recognition rate: 0.7570810398377623
Mean relatedness of consumed Cs: 0.004054144162165778
Var of relatedness of consumed Cs: 1.3161985026946234E-4

Stats for Artefact Typ_D_0:

Artefact type: 4

Total active steps: 5000
Total number of contexts consumed: 0
Total number of contexts produced: 5000
Overall recognition rate: 0.870008390929034
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_E_0:

Artefact type: 5

Total active steps: 5000
Total number of contexts consumed: 9996

Total number of contexts produced: 4998
Overall recognition rate: 0.6587171862942044
Mean relatedness of consumed Cs: 0.0038815526210484274
Var of relatedness of consumed Cs: 1.3035111983556683E-4

General Context Stats

Total number of contexts consumed: 19994
Total number of contexts produced: 24997
Mean derive time: 1.4999499849955051
Max internal context reliability: 0.9996547197270061
Min internal context reliability: 0.44634731740350775
Mean internal context reliability: 0.8052339051652415

Simulation stats

Total number of simulation steps	5000
Total number of simTicks:	5000
Total number of artefacts:	5
Total number of contexts in CSpace:	5
Total number of derivations:	24997.0
Mean number of derivations per artefact:	4999.4
Max recognition rate:	0.8712649900538805
Min recognition rate:	0.6587171862942044
Mean recognition rate:	0.805
Var of recognition rate:	0.009102896416529276
Min Seed Reliability:	0.691505594219769
Max Seed Reliability:	0.9996547197270061

A.3.2 FCR mit Filter

List of Artefacts

1 Artefact Typ_A_0 of type 1
2 Artefact Typ_B_0 of type 2
3 Artefact Typ_C_0 of type 3
4 Artefact Typ_D_0 of type 4
5 Artefact Typ_E_0 of type 5

Stats for Artefact Typ_A_0:

Artefact type: 1

Total active steps: 5000

Anhang A Simulationsergebnisse

Total number of contexts consumed: 0
Total number of contexts produced: 5000
Overall recognition rate: 0.8689201073928196
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_B_0:

Artefact type: 2

Total active steps: 5000
Total number of contexts consumed: 0
Total number of contexts produced: 5000
Overall recognition rate: 0.869539019416033
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_C_0:

Artefact type: 3

Total active steps: 5000
Total number of contexts consumed: 740
Total number of contexts produced: 370
Overall recognition rate: 0.864982386319853
Mean relatedness of consumed Cs: 0.0035135135135135115
Var of relatedness of consumed Cs: 1.231231231231232E-4

Stats for Artefact Typ_D_0:

Artefact type: 4

Total active steps: 5000
Total number of contexts consumed: 0
Total number of contexts produced: 5000
Overall recognition rate: 0.8706666221658353
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_E_0:

Artefact type: 5

Total active steps: 5000
Total number of contexts consumed: 208
Total number of contexts produced: 104
Overall recognition rate: 0.8054049075216244
Mean relatedness of consumed Cs: 0.0035256410256410253

Var of relatedness of consumed Cs: 1.0611152601443866E-4

General Context Stats

Total number of contexts consumed: 948
Total number of contexts produced: 15474
Mean derive time: 1.2194092827004224
Max internal context reliability: 0.999948543733433
Min internal context reliability: 0.6780530165483398
Mean internal context reliability: 0.8691633926711935

Simulation stats

Total number of simulation steps	5000
Total number of simTicks:	5000
Total number of artefacts:	5
Total number of contexts in CSpace:	3
Total number of derivations:	15474.0
Mean number of derivations per artefact:	3094.8
Max recognition rate:	0.8706666221658353
Min recognition rate:	0.8054049075216244
Mean recognition rate:	0.856
Var of recognition rate:	8.014608122449422E-4
Min Seed Reliability:	0.6780530165483398
Max Seed Reliability:	0.999948543733433

A.4 Simulation: GRC

A.4.1 GRC Auffächerung und Multiplikation ohne Filter

List of Artefacts

1 Artefact Typ_A_0 of type 1
2 Artefact Typ_B_0 of type 2
3 Artefact Typ_B_1 of type 2
4 Artefact Typ_B_2 of type 2
5 Artefact Typ_C_0 of type 3

Stats for Artefact Typ_A_0:

Artefact type: 1

Anhang A Simulationsergebnisse

Total active steps: 5000
Total number of contexts consumed: 0
Total number of contexts produced: 5000
Overall recognition rate: 0.8688415570450692
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_B_0:

Artefact type: 2

Total active steps: 5000
Total number of contexts consumed: 4999
Total number of contexts produced: 4999
Overall recognition rate: 0.8672104273207696
Mean relatedness of consumed Cs: 0.23825431753017265
Var of relatedness of consumed Cs: 0.17862857352049336

Stats for Artefact Typ_B_1:

Artefact type: 2

Total active steps: 5000
Total number of contexts consumed: 4999
Total number of contexts produced: 4999
Overall recognition rate: 0.8672104273207696
Mean relatedness of consumed Cs: 0.2382409815296394
Var of relatedness of consumed Cs: 0.17863448471117863

Stats for Artefact Typ_B_2:

Artefact type: 2

Total active steps: 5000
Total number of contexts consumed: 4999
Total number of contexts produced: 4999
Overall recognition rate: 0.8672104273207696
Mean relatedness of consumed Cs: 0.23825431753017273
Var of relatedness of consumed Cs: 0.1786285735204932

Stats for Artefact Typ_C_0:

Artefact type: 3

Total active steps: 5000
Total number of contexts consumed: 14994
Total number of contexts produced: 4998
Overall recognition rate: 0.6075336458767863

Mean relatedness of consumed Cs: 0.3868458494508915
Var of relatedness of consumed Cs: 0.08781520814236893

General Context Stats

Total number of contexts consumed: 29991
Total number of contexts produced: 24995
Mean derive time: 2.708977244311089
Max internal context reliability: 1.0878
Min internal context reliability: 0.0699006781590027
Mean internal context reliability: 0.8156117513841601

Simulation stats

Total number of simulation steps	5000
Total number of simTicks:	5000
Total number of artefacts:	5
Total number of contexts in CSpace:	15
Total number of derivations:	24995.0
Mean number of derivations per artefact:	4999.0
Max recognition rate:	0.8688415570450692
Min recognition rate:	0.6075336458767863
Mean recognition rate:	0.816
Var of recognition rate:	0.013529294932749134
Min Seed Reliability:	0.6716816209088418
Max Seed Reliability:	0.9999512081848899

A.4.2 GRC Auffächerung und Multiplikation mit Filter

List of Artefacts

1 Artefact Typ_A_0 of type 1
2 Artefact Typ_B_0 of type 2
3 Artefact Typ_B_1 of type 2
4 Artefact Typ_B_2 of type 2
5 Artefact Typ_C_0 of type 3

Stats for Artefact Typ_A_0:

Artefact type: 1

Total active steps:	5000
Total number of contexts consumed:	0
Total number of contexts produced:	5000

Anhang A Simulationsergebnisse

Overall recognition rate: 0.8703143415357706
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

Stats for Artefact Typ_B_0:

Artefact type: 2

Total active steps: 5000
Total number of contexts consumed: 4999
Total number of contexts produced: 4999
Overall recognition rate: 0.8709911831715688
Mean relatedness of consumed Cs: 0.2380942855237707
Var of relatedness of consumed Cs: 0.17869948432797092

Stats for Artefact Typ_B_1:

Artefact type: 2

Total active steps: 5000
Total number of contexts consumed: 4999
Total number of contexts produced: 4999
Overall recognition rate: 0.8709911831715688
Mean relatedness of consumed Cs: 0.23811428952457112
Var of relatedness of consumed Cs: 0.17869062327871646

Stats for Artefact Typ_B_2:

Artefact type: 2

Total active steps: 5000
Total number of contexts consumed: 4999
Total number of contexts produced: 4999
Overall recognition rate: 0.8709911831715688
Mean relatedness of consumed Cs: 0.23811428952457114
Var of relatedness of consumed Cs: 0.17869062327871657

Stats for Artefact Typ_C_0:

Artefact type: 3

Total active steps: 5000
Total number of contexts consumed: 7386
Total number of contexts produced: 2462
Overall recognition rate: 0.7584927144079863
Mean relatedness of consumed Cs: 0.00375485152089539
Var of relatedness of consumed Cs: 4.0876524052520745E-5

General Context Stats

Total number of contexts consumed: 22383
Total number of contexts produced: 22459
Mean derive time: 2.468927200870629
Max internal context reliability: 1.069167608619676
Min internal context reliability: 0.4807111894237882
Mean internal context reliability: 0.8585081946914551

Simulation stats

Total number of simulation steps 5000
Total number of simTicks: 5000
Total number of artefacts: 5
Total number of contexts in CSpace: 13
Total number of derivations: 22459.0
Mean number of derivations per artefact: 4491.8
Max recognition rate: 0.8709911831715688
Min recognition rate: 0.7584927144079863
Mean recognition rate: 0.848
Var of recognition rate: 0.0025236583529878696
Min Seed Reliability: 0.6817264707247142
Max Seed Reliability: 0.9999755769860719

A.4.3 GRC zyklische Verarbeitung ohne Filter

List of Artefacts

1 Artefact Typ_A_0 of type 1
2 Artefact Typ_B_0 of type 2
3 Artefact Typ_C_0 of type 3
4 Artefact Typ_D_0 of type 4

Stats for Artefact Typ_A_0:

Artefact type: 1

Total active steps: 5000
Total number of contexts consumed: 4769
Total number of contexts produced: 4769
Overall recognition rate: 0.5091523750013758
Mean relatedness of consumed Cs: 0.807989096246593
Var of relatedness of consumed Cs: 0.15491169227481885

Anhang A Simulationsergebnisse

Stats for Artefact Typ_B_0:

Artefact type: 2

Total active steps: 5000
Total number of contexts consumed: 4768
Total number of contexts produced: 4768
Overall recognition rate: 0.5087213099338026
Mean relatedness of consumed Cs: 0.8096196868008936
Var of relatedness of consumed Cs: 0.15391112493971515

Stats for Artefact Typ_C_0:

Artefact type: 3

Total active steps: 5000
Total number of contexts consumed: 4767
Total number of contexts produced: 4767
Overall recognition rate: 0.5081804581474955
Mean relatedness of consumed Cs: 0.8095797496678541
Var of relatedness of consumed Cs: 0.15393581206897283

Stats for Artefact Typ_D_0:

Artefact type: 4

Total active steps: 41
Total number of contexts consumed: 0
Total number of contexts produced: 41
Overall recognition rate: 0.8859075828474393
Mean relatedness of consumed Cs: NaN
Var of relatedness of consumed Cs: NaN

General Context Stats

Total number of contexts consumed: 14304
Total number of contexts produced: 14345
Mean derive time: 683.3587108501164
Max internal context reliability: 0.999
Min internal context reliability: 0.1288437726973299
Mean internal context reliability: 0.5097629374020071

Simulation stats

Total number of simulation steps 5000
Total number of simTicks: 5000

```
Total number of artefacts:      4
Total number of contexts in CSpace: 3
Total number of derivations:    14345.0
Mean number of derivations per artefact: 3586.25
Max recognition rate:           0.8859075828474393
Min recognition rate:           0.5081804581474955
Mean recognition rate:          0.603
Var of recognition rate:        0.03557443123398467
Min Seed Reliability:           0.7909727889007262
Max Seed Reliability:           0.999
```

A.4.4 GRC zyklische Verarbeitung mit Filter

List of Artefacts

```
1 Artefact Typ_A_0 of type 1
2 Artefact Typ_B_0 of type 2
3 Artefact Typ_C_0 of type 3
4 Artefact Typ_D_0 of type 4
```

Stats for Artefact Typ_A_0:

Artefact type: 1

```
Total active steps:      5000
Total number of contexts consumed: 56
Total number of contexts produced: 56
Overall recognition rate: 0.8666918567244027
Mean relatedness of consumed Cs: 0.018452380952380956
Var of relatedness of consumed Cs: 0.0178553391053391
```

Stats for Artefact Typ_B_0:

Artefact type: 2

```
Total active steps:      5000
Total number of contexts consumed: 56
Total number of contexts produced: 56
Overall recognition rate: 0.865825164867678
Mean relatedness of consumed Cs: 0.018452380952380956
Var of relatedness of consumed Cs: 0.0178553391053391
```

Stats for Artefact Typ_C_0:

Anhang A Simulationsergebnisse

Artefact type: 3

Total active steps:	5000
Total number of contexts consumed:	56
Total number of contexts produced:	56
Overall recognition rate:	0.8649593397028101
Mean relatedness of consumed Cs:	0.018452380952380956
Var of relatedness of consumed Cs:	0.0178553391053391

Stats for Artefact Typ_D_0:

Artefact type: 4

Total active steps:	56
Total number of contexts consumed:	0
Total number of contexts produced:	56
Overall recognition rate:	0.8675594161405432
Mean relatedness of consumed Cs:	NaN
Var of relatedness of consumed Cs:	NaN

General Context Stats

Total number of contexts consumed:	168
Total number of contexts produced:	224
Mean derive time:	1.9999999999999993
Max internal context reliability:	0.9756014921103677
Min internal context reliability:	0.7497204081277599
Mean internal context reliability:	0.866258944358858

Simulation stats

Total number of simulation steps	5000
Total number of simTicks:	5000
Total number of artefacts:	4
Total number of contexts in CSpace:	0
Total number of derivations:	224.0
Mean number of derivations per artefact:	56.0
Max recognition rate:	0.8675594161405432
Min recognition rate:	0.8649593397028101
Mean recognition rate:	0.866
Var of recognition rate:	1.2519256267302688E-6
Min Seed Reliability:	0.7519740751830576
Max Seed Reliability:	0.9756014921103677

A.5 Simulation: Gesamtszenario

A.5.1 Zusammenfassung der Simulationsergebnisse ohne Filter

General Context Stats

Total number of contexts consumed: 407981
Total number of contexts produced: 226315
Mean derive time: 1899.9427448512022
Max internal context reliability: 0.9999776983888481
Min internal context reliability: 2.1982562611506347E-32
Mean internal context reliability: 0.36708635294181824

Simulation stats

Total number of simulation steps: 5000
Total number of simTicks: 5000
Total number of artefacts: 118
Total number of contexts in CSpace: 351
Total number of derivations: 226315.0
Mean number of derivations per artefact: 1917.9237288135594
Max recognition rate: 0.8728257168509568
Min recognition rate: 0.0031009438815747395
Mean recognition rate: 0.54
Var of recognition rate: 0.16688781859932586
Min Seed Reliability: 0.6432604819755752
Max Seed Reliability: 0.9999776983888481

A.5.2 Zusammenfassung der Simulationsergebnisse mit Filter

General Context Stats

Total number of contexts consumed: 14159
Total number of contexts produced: 96236
Mean derive time: 1.7544650040225318
Max internal context reliability: 0.9998957349982539
Min internal context reliability: 0.5258489688069505
Mean internal context reliability: 0.8642177999150117

Simulation stats

Total number of simulation steps	5000
Total number of simTicks:	5000
Total number of artefacts:	118
Total number of contexts in CSpace:	146
Total number of derivations:	96236.0
Mean number of derivations per artefact:	815.5593220338983
Max recognition rate:	0.9100865972732224
Min recognition rate:	0.6140801078979289
Mean recognition rate:	0.801
Var of recognition rate:	0.00924636958647103
Min Seed Reliability:	0.6650447443989842
Max Seed Reliability:	0.9998957349982539

Anhang B

Studie: Zählung der Gegenstände in Büroumgebungen

Dieser Anhang enthält die Ergebnisse einer Studie mit dem Ziel eine Datenbasis zu schaffen welche es erlaubt, die Anzahl der computerisierten Gegenständen und Ubiquitous Computing Artefakten in typischen zukünftigen Büroumgebungen abzuschätzen. Zu diesem Zweck wurden in verschiedenen Büros alle vorhandenen Gegenstände gezählt und katalogisiert.

Diese Daten bilden die Grundlage für die Schätzung der Anzahl computerisierter Gegenstände in zukünftigen ubiquitären Informationsumgebungen. Da unterschiedliche Annahmen über die Art der potentiellen zukünftigen Artefakte zu unterschiedlichen Ergebnissen führen, werden hier auch die reinen aufbereitenden Zählungsergebnisse dokumentiert. Diese können so auch in späteren Arbeiten als Datenbasis für ähnliche Überlegungen genutzt werden. Daneben werden die Artefakte herausgehoben, für welche im Rahmen dieser Arbeit angenommen wurde, dass sie potentiell in Zukunft computerisiert sein werden.

Die Zählungsprotokolle sind wie folgt aufgebaut: Im Kopfteil befinden sich Daten über Zeitpunkt, Ort und Dauer der Zählung. Die erste Spalte der Tabelle enthält den Namen des Zählformulars. In der zweiten Spalte sind die bei der Zählung gefundenen Gegenstände verzeichnet. Spalte drei enthält die Anzahl dieser Gegenstände. Bei dieser Zahl handelt es sich um die oben angesprochenen Basisdaten der Zählung. Die vierte Spalte enthält die Anzahl an Gegenständen, welche in Zukunft potentiell computerisiert sein könnten. In der letzten Spalte ist der Ort, an dem die Gegenstände gefunden wurden, näher spezifiziert.

B.1 Bürozahlung: A. Krohn

Datum	08.10.04			
Ort	KA, Teco, Büro Krohn, R15			
Zähler, Name	Michael Demel			
Nr, ID	081004_TECO_R15_MD			
Zeit- auf- wand	3h			

Form	Gegenstand	Anz.	Anz.	Ort
1/9	Schreibtische	2	2	Rolltisch
Haupt- form	Rollcontainer	0		Wand, Eingang
	Tische	3	3	
	Schränke	0		
	Schreibtischstühle	1	1	
	Regale	1	1	
	Türen	1	1	
	Wände (freier Platz)	3	3	
	Pflanzen	1	1	
	Fenster	2	2	
	Papierkörbe	1	1	
	Lampen	2	2	
2/9	Schubladen	0		
Detail- form	Aktenablagen (Kunststoff)	0		
	Ordner	0		
	Kartons	0		
	Bücher	0		
	Briefe	0		
	Zeitschriften	0		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	lose Blätter	0		
	Notizen	0		
	geheftete Stapel	0		
	Kulis / Bleistifte / Stabilos	0		
	Marker / Eddings	0		
	andere Stifte	0		
	Boardmarker	0		
	Zuleitungen	0		
	Datenkabel	0		
	CD / DVD / Speichermedien	0		
	Verlängerungen + Steckdo- senleisten	1		
	Taschen	0		
+	Monitor	1	1	
3/9	Schubladen	0		Regal_Wand
Detail- form	Aktenablagen (Kunststoff)	2	2	Wand, Eingang
	Ordner	2	2	
	Kartons	5		
	Bücher	21	21	
	Briefe	10		
	Zeitschriften	64		
	lose Blätter	51		
	Notizen	0		
	geheftete Stapel	25		
	Kulis / Bleistifte / Stabilos	0		
	Marker / Eddings	0		
	andere Stifte	0		
	Boardmarker	0		
	Zuleitungen	0		
	Datenkabel	0		
	CD / DVD / Speichermedien	1		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	Verlängerungen + Steckdosensleisten	1		
	Taschen	2	2	
3.1/9	Regenschirm	3	3	
Zusatzform	Ventilatoren	3	3	
	Trafo	1		
	Platinen	3		
	Leuchtstofflampen	1		
	Kalenderposter	1		
	Winkelmesser	1		
	Tee + Zucker	27		
	Klopapierrolle	1		
	Gehäuse, Kunststoff	3		
	Flaschen	23		
	Kaugummi, Bonbon Pckg.	3		
	Tragegurt für Tasche	1		
4/9				
Detailform	Schubladen	0		Tisch hinten
	Aktenablagen (Kunststoff)	2	2	großer Tisch hinten
	Ordner	0		
	Kartons	6		
	Bücher	0		
	Briefe	0		
	Zeitschriften	0		
	lose Blätter	10		
	Notizen	0		
	geheftete Stapel	4		
	Kulis / Bleistifte / Stabilos	1	1	
	Marker / Eddings	1	1	
	andere Stifte	0		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	Boardmarker	0		
	Zuleitungen	0		
	Datenkabel	1		
	CD / DVD / Speichermedien	9		
	Verlängerungen + Steckdos- enleisten	0		
	Taschen	1	1	
4.1/9	PC Tower	2	2	
Zusatz- form	Schere	1	1	
	Verpackungsmaterial	19		
	Flaschen	1		
	Tee + Espresso + Zucker	3		
	Trinkglas	1		
	Löffel	1		
	Platinen	35		
	Netzgeräte	3		
	Aufbewahrung, Kunststoff- kiste	2	2	
	Kameramodul	1		
	Touchscreen, Display	1	1	
	Batterie, Sonderform	1		
5/9	Schubladen	0		Wände Boden
Detail- form	Aktenablagen (Kunststoff)	0		alle Wände + Bo- den
	Ordner	0		
	Kartons	0		
	Bücher	0		
	Briefe	0		
	Zeitschriften	0		
	lose Blätter	0		
	Notizen	7		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	geheftete Stapel	0		
	Kulis / Bleistifte / Stabilos	0		
	Marker / Eddings	0		
	andere Stifte	0		
	Boardmarker	0		
	Zuleitungen	1		
	Datenkabel	6		
	CD / DVD / Speichermedien	0		
	Verlängerungen + Steckdo- senleisten	2		
	Taschen	0		
+	Pinnwand	1	1	
	Vorhang	2		
	Rauch- + Bewegungsmelder	2	2	
	Poster	2		
	Pinns, Magnet	9		
	Sandalen, Paar	1		
6/9	Schubladen	0		Hauptschreibtisch
Detail- form	Aktenablagen (Kunststoff)	0		Ecke hinten, Fens- ter
	Ordner	0		
	Kartons	0		
	Bücher	2	2	
	Briefe	2		
	Zeitschriften	0		
	lose Blätter	43		
	Notizen	7		
	geheftete Stapel	3		
	Kulis / Bleistifte / Stabilos	5	5	
	Marker / Eddings	1	1	
	andere Stifte	0		
	Boardmarker	0		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	Zuleitungen	3		
	Datenkabel	12		
	CD / DVD / Speichermedien	0		
	Verlängerungen + Steckdosenleisten	0		
	Taschen	0		
6.1/9	PC Tower	1	1	
Zusatzform	PC Lautsprecher	2	2	
	Monitor	1	1	
	PC Maus	1	1	
	Taschenrechner	1	1	
	Gehäuse, Kunststoff	1	1	
	Labello	1		
	Platinen	5		
	Bauteile, Elektro	5		
	Akku AAA	1		
	Figur, Metall, klein	1		
	Telefon	1	1	
	Tesa-Klebeband	2	2	
	Holzkistchen	1	1	
	Flaschendeckel	1		
7/9	Schubladen	0		Tisch klein Mitte
Detailform	Aktenablagen (Kunststoff)	1	1	Mitte Fenster
	Ordner	0		
	Kartons	2		
	Bücher	3	3	
	Briefe	0		
	Zeitschriften	0		
	lose Blätter	3		
	Notizen	1		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	geheftete Stapel	1		
	Kulis / Bleistifte / Stabilos	0		
	Marker / Eddings	0		
	andere Stifte	0		
	Boardmarker	0		
	Zuleitungen	0		
	Datenkabel	5		
	CD / DVD / Speichermedien	0		
	Verlängerungen + Steckdosensleisten	0		
	Taschen	0		
7.1/9	Hub / Switch	2		
Zusatzform	Ventilatoren	1	1	
	Untertopf (Blumen)	1		
	Deckel, Kunststoff, klein	1		
	Locher	1	1	
	Hefter	1	1	
	Mousepad	1		
	Platinen	6		
	Akku-Halterungen	6		
	Visitenkarten	480		
	Clipboard	1		
	Audio-Level-Messgerät	1		
8/9	Schubladen	7	7	Tisch Mitte groß
Detailform	Aktenablagen (Kunststoff)	4	4	Schreibtisch, Mitte
	Ordner	0		
	Kartons	2		
	Bücher	0		
	Briefe	2		
	Zeitschriften	1		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	lose Blätter	9		
	Notizen	2		
	geheftete Stapel	21		
	Kulis / Bleistifte / Stabilos	0		
	Marker / Eddings	3	3	
	andere Stifte	0		
	Boardmarker	0		
	Zuleitungen	0		
	Datenkabel	1		
	CD / DVD / Speichermedien	0		
	Verlängerungen + Steckdo- senleisten	0		
	Taschen	0		
8.1/9	Platinen	51		
Zusatz- form	Bauteile, Elektro	6		
	Schere	1	1	
	Poster, gerollt	8		
	Gehäuse, Kunststoff	8		
	9V-Block-Batterie	1		
	Verpackungsmaterial	8		
	Akku-Halterungen AAA	28		
	Teebeutel	1		
	KVV-Fahrplan	1		
	Werkzeugkasten, klein	1	1	
9/9	Schubladen	0		Schubladentisch Mitte
Detail- form	Aktenablagen (Kunststoff)	0		
	Ordner	0		
	Kartons	7		
	Bücher	0		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	Briefe	0		
	Zeitschriften	0		
	lose Blätter	30		
	Notizen	24		
	geheftete Stapel	0		
	Kulis / Bleistifte / Stabilos	10	10	
	Marker / Eddings	8	8	
	andere Stifte	7	7	
	Boardmarker	5	5	
	Zuleitungen	0		
	Datenkabel	2		
	CD / DVD / Speichermedien	1		
	Verlängerungen + Steckdo- senleisten	3		
	Taschen	0		
9.1/9	Hängeakten, Papier	27		
Zusatz- form	Klebstoff, Tube	1		
	Klebepunkte Pckg.	2		
	Schlüssel	6	6	
	Münzen	22		
	Tabletten Pckg.	1		
	Heftklammern Pckg.	2		
	Steckeradapter	1		
	Bauteile, Elektro	22		
	Platinen	21		
	Gehäuse, Kunststoff	1		
	Büroklammern	16		
	Pinns	8		
	Wäscheklammern	2		
	Akku, utopisch	1		
	Prospekthüllen	4		
	Bleispitzer	1		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	Trennwand, Schublade	1		
	Tastatur	1	1	
	Paketschnur, Rolle	2		
	Gesamt / Relevant	1439	144	

B.2 Bürozahlung: P. Spieß

Datum	09.12.04			
Ort	KA, SAP, Büro Spieß, R5			
Zähler, Name	Michael Demel			
Nr, ID	091204_TECO_R5_MD_PS			
Zeit-auf-wand	1,5h			
Form	Gegenstand	Anz.	Anz.	Ort
1/5	Schreibtische	3	3	
Haupt-form	Rollcontainer	3	3	
	Tische	1	1	
	Schränke	1	1	
	Schreibtischstühle	2	2	
	Regale	0	0	
	Türen	1	1	
	Wände (freier Platz)	2	2	
	Pflanzen	1	1	
	Fenster	2	2	
	Papierkörbe	1	1	
	Lampen	2	2	
2/5	Schubladen	0		Boden
Detail-form	Aktenablagen (Kunststoff)	0		alles Boden
	Ordner	0		
	Kartons	4		
	Bücher	0		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	Briefe	2		
	Zeitschriften	0		
	lose Blätter	1		
	Notizen	0		
	geheftete Stapel	1		
	Kulis / Bleistifte / Stabilos	0		
	Marker / Eddings	0		
	andere Stifte	0		
	Boardmarker	0		
	Zuleitungen	6		
	Datenkabel	3		
	CD / DVD / Speichermedien	0		
	Verlängerungen + Steckdo- senleisten	3		
	Taschen	3	3	
+	Sprudelkästen	1	1	
+	Schlüssel	2	2	
+	Standventilator	1	1	
3/5	Schubladen	0		Wand links
Detail- form	Aktenablagen (Kunststoff)	0		große Wand, linke Seite
	Ordner	0		
	Kartons	0		
	Bücher	0		
	Briefe	0		
	Zeitschriften	0		
	lose Blätter	4		
	Notizen	3		
	geheftete Stapel	0		
	Kulis / Bleistifte / Stabilos	0		
	Marker / Eddings	0		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	andere Stifte	0		
	Boardmarker	0		
	Zuleitungen	0		
	Datenkabel	0		
	CD / DVD / Speichermedien	0		
	Verlängerungen + Steckdosenleisten	0		
	Taschen	0		
+	Posterkalender	2		
+	Pinnboard	1	1	
+	Pinns	45		
4/5	Schubladen	8	8	Schreibtisch
Detail-form	Aktenablagen (Kunststoff)	0		Schreibtisch + Rollcontainer
	Ordner	0		
	Kartons	0		
	Bücher	1	1	
	Briefe	3		
	Zeitschriften	0		
	lose Blätter	43		
	Notizen	1		
	geheftete Stapel	4		
	Kulis / Bleistifte / Stabilos	2	2	
	Marker / Eddings	1	1	
	andere Stifte	1	1	
	Boardmarker	0		
	Zuleitungen	3		
	Datenkabel	10		
	CD / DVD / Speichermedien	10		
	Verlängerungen + Steckdosenleisten	2		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	Taschen	0		
4.1/5	Gummiband	2		
Zusatz- form	Bridges + Particles	2		
	Steckernetzteil	3		
	Akku AAA	4		
	Telefon	1	1	
	Magnesium Tabletten Pckg.	1		
	Laptop	2	2	
	PET-Flaschen	2		
	Kopfhörer	1		
	Tassen	2	2	
	LAN Switch	1		
	Wechselfestplatte	1	1	
	Monitor	1	1	
	Tastatur	1	1	
	Maus	1	1	
	Jacke	1	1	
	Handy	1	1	
	Schlüssel	1	1	
	Post-It Pckg.	1		
	Kabelbinder	2		
	Hefter	1	1	
	Kürbiskern Pckg.	1		
	Klopapierrolle	1		
	Codekarte	1		
	Plastiktüte	1		
5/5	Schubladen	0		Schrank
Detail- form	Aktenablagen (Kunststoff)	0		Schrank, Eingang links
	Ordner	0		
	Kartons	2		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	Bücher	0		
	Briefe	0		
	Zeitschriften	0		
	lose Blätter	0		
	Notizen	0		
	geheftete Stapel	1		
	Kulis / Bleistifte / Stabilos	0		
	Marker / Eddings	0		
	andere Stifte	0		
	Boardmarker	0		
	Zuleitungen	0		
	Datenkabel	0		
	CD / DVD / Speichermedien	5		
	Verlängerungen + Steckdo- senleisten	1		
	Taschen	1		
5.1/5	Styroporverpackung	2		
Zusatz- form	Golfball	1		
	Backsteine	3		
	Briefwaage	1	1	
	Motherboard	1		
	Gesamt / Relevant	244	54	

B.3 Bürozahlung: T. Zimmer

Datum	02.11.04			
Ort	KA, Teco, Büro Zimmer, R13			
Zähler, Name	Michael Demel			
Nr, ID	021104_TECO_R13_MD			
Zeit-auf-wand	3h			
Form	Gegenstand	Anz.	Anz.	Ort
1/6	Schreibtische	2	2	
Haupt-form	Rollcontainer	2	2	
	Tische	1	1	
	Schränke	1	1	
	Schreibtischstühle	1	1	
	Regale	1	1	
	Türen	1	1	
	Wände (freier Platz)	1	1	
	Pflanzen	3	3	
	Fenster	2	2	
	Papierkörbe	2	2	
	Lampen	2	2	
2/6	Schubladen	0		Wand Boden
Detail-form	Aktenablagen (Kunststoff)	0		rechte Wand + Boden alles
	Ordner	1	1	
	Kartons	3		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	Bücher	0		
	Briefe	0		
	Zeitschriften	0		
	lose Blätter	26		
	Notizen	6		
	geheftete Stapel	1		
	Kulis / Bleistifte / Stabilos	0		
	Marker / Eddings	0		
	andere Stifte	0		
	Boardmarker	1	1	
	Zuleitungen	11		
	Datenkabel	19		
	CD / DVD / Speichermedien	4		
	Verlängerungen + Steckdo- senleisten	2		
	Taschen	0		
+	Sprudelkästen	2	2	
2.1/6	Lochbrett ca. 1m	1		
Zusatz- form	Ventilator	1	1	
	Netzteil	1		
	Paketfüllmaterial (Karton)	1		
	Regenschirm	2	2	
	Dartspiel	1		
	PET-Flasche	1		
	Kalenderposter	6		
	Tesaabroller	1	1	
	Teelicht	1		
	Platinen	1		
	Pinnwand	2	2	
	Lichtschalter	1		
	Befestigungswinkel	2		
	Visitenkarten	1		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	Pinns	62		
3/6	Schubladen	0		Schrank
Detail- form	Aktenablagen (Kunststoff)	1	1	Eingang rechts
	Ordner	34	34	
	Kartons	23		
	Bücher	15	15	
	Briefe	1		
	Zeitschriften	72		
	lose Blätter	5		
	Notizen	9		
	geheftete Stapel	33		
	Kulis / Bleistifte / Stabilos	1	1	
	Marker / Eddings	2	2	
	andere Stifte	0		
	Boardmarker	1	1	
	Zuleitungen	2		
	Datenkabel	18		
	CD / DVD / Speichermedien	8		
	Verlängerungen + Steckdo- senleisten	0		
	Taschen	2	2	
+	Platinen	296		
3.1/6	Tastatur	2	2	
Zusatz- form	Maus	1	1	
	Wandfarbe	1		
	Blumenerde	1		
	Ferngesteuerte Autos + Fernbed.	4		
	PC-Steckkarten	3		
	Antennen	2		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	CD-Hüllen	2		
	Handy	3	3	
	Digicam	2	2	
	Organizer (PDA)	1	1	
	Multimedia Player	1	1	
	Schlüssel	1	1	
	Steckeradapter	3		
4/6	Schubladen	0		Regal
Detail- form	Aktenablagen (Kunststoff)	2	2	Eingang rechts Re- gal
	Ordner	11	11	
	Kartons	7		
	Bücher	7	7	
	Briefe	14		
	Zeitschriften	22		
	lose Blätter	26		
	Notizen	3		
	geheftete Stapel	27		
	Kulis / Bleistifte / Stabilos	1	1	
	Marker / Eddings	1	1	
	andere Stifte	0		
	Boardmarker	3	3	
	Zuleitungen	6		
	Datenkabel	8		
	CD / DVD / Speichermedien	2		
	Verlängerungen + Steckdo- senleisten	0		
	Taschen	2	2	
+	elektronische Bauteile	1000		
4.1/6	Glasflaschen	8		
Zusatz- form	PET-Flasche	1		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	Bilderrahmen	2	2	
	Aufbewahrung / Tütchen	80		
	Briefumschlag (leer)	71		
	Türpaneel	1		
	Steckernetzteil	4		
	Digicamcorderzubehör	1		
	T-Shirts	5		
	Conrad-Power-Manager	1		
	Integrated-Home-Controller	1		
	Ordnerleinlagen (Trennblätter)	50		
	Overheadfolien	100		
	Heftklammern Pckg.	1		
	Thermodrucker	1		
	Gabelschlüssel	1		
	Papier (weiß)	100		
	Funkfernsteuerung	1		
	Klebeband	11		
	Schrauben	3		
	Winkel	1		
	Funkempfängerschalter	4		
	Thermodruckerbänder	13		
	Pinns Pckg.	1		
	CD-Hüllen	1		
	Dachträger Auto	1		
	Puppenhaus + Zubehör	1		
	Halogenlampen	2		
	Schrittmotor	1		
	Platinen	5		
	Schlüssel	2	2	
5/6	Schubladen	7	7	Schreibtisch
Fortsetzung nächste Seite ...				

Fortsetzung ...				
Detail- form	Aktenablagen (Kunststoff)	0		Hauptschreibtisch, Fenster
	Ordner	1	1	
	Kartons	8		
	Bücher	2	2	
	Briefe	0		
	Zeitschriften	0		
	lose Blätter	27		
	Notizen	3		
	geheftete Stapel	3		
	Kulis / Bleistifte / Stabilos	14	14	
	Marker / Eddings	14	14	
	andere Stifte	7	7	
	Boardmarker	2	2	
	Zuleitungen	1		
	Datenkabel	2		
	CD / DVD / Speichermedien	8		
	Verlängerungen + Steckdo- senleisten	0		
	Taschen	0		
+	Ladestation (PDA)	4		
	Platinen	10		
5.1/6	Schlüssel	91		
Zusatz- form	Folie	7		
	Taschentücher Pckg.	3		
	Tee Pckg.	2		
	Deo	1		
	Brillenputztücher Pckg.	1		
	Handcreme	2		
	Notizbuch	1		
	Labello	1		
	Brausetabletten Pckg.	2		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	Schokoriegel	3		
	Pflaster Pckg.	1		
	Müslischüssel + Müsli	1		
	Betriebsanleitung	2		
	Buntlackdose	1		
	elektronische Bauteile	422		
	Hängeakten, Papier	19		
	Visitenkarten	230		
	PDA-Stift	2		
	Bleistiftminen Pckg.	1		
	Heftklammern Pckg.	2		
	Büroklammern	24		
	Kabelbinder	2		
	Laptop-Stick-Ersatz	1		
	Radiergummi	4		
	Hefter	1	1	
	Locher	1	1	
	Post-It Pckg.	7		
	Taschenrechner	1	1	
	Tipp-Ex	1		
	Klebestift	1		
	Schreibtischunterlage	1	1	
	Steckernetzteil	4		
	GPS-Modul	1		
	Visitenkartenpapier Pckg.	1		
	Papier Pckg.	1		
	Heftklammernentferner	1		
	Bleispitzer	1		
	Ausweis	1		
	Schlüsselringe	18		
	Geldtruhe / Safe	1		
	Organizer (PDA)	1	1	
Fortsetzung nächste Seite ...				

Fortsetzung ...				
6/6	Schubladen	0		Tische Mitte
Detail- form	Aktenablagen (Kunststoff)	5	5	Tische Mitte, klein und groß
	Ordner	0		
	Kartons	1		
	Bücher	1	1	
	Briefe	3		
	Zeitschriften	1		
	lose Blätter	31		
	Notizen	10		
	geheftete Stapel	10		
	Kulis / Bleistifte / Stabilos	1	1	
	Marker / Eddings	1	1	
	andere Stifte	0		
	Boardmarker	0		
	Zuleitungen	2		
	Datenkabel	5		
	CD / DVD / Speichermedien	3		
	Verlängerungen + Steckdo- senleisten	1		
	Taschen	1	1	
	Monitor	1	1	
	Telefon	1	1	
6.1/6	Post-It Pckg.	3		
Zusatz- form	Clipboard	1		
	Taschenrechner	1	1	
	PC-Lautsprecher	3		
	IR-Empfänger	1		
	Platinen	2		
	Knopfzelle	1		
	Festplatte	1		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	Modellauto	1		
	Tupper-Deckel	1		
	Bleigelakku	1		
	PDA-Stift	1		
	Maus	1		
	Tastatur	1		
	PC-Miditower	1		
	Prospekthüllen	1		
	Kopfhörer	1	1	204
	Gesamt / Relevant	3429	192	

B.4 Bürozahlung: C. Decker

Datum	02.11.04			
Ort	KA, Teco, Büro Decker, R13			
Zähler, Name	Michael Demel			
Nr, ID	021104_TECO_R13_MD			
Zeit-auf-wand	3h			
Form	Gegenstand	Anz.	Anz.	Ort
1/6	Schreibtische	2	2	
Haupt-form	Rollcontainer	1	1	
	Tische	2	2	
	Schränke	2	1	
	Schreibtischstühle	2	2	
	Regale	0	0	
	Türen	1	1	
	Wände (freier Platz)	2	2	
	Pflanzen	6	0	
	Fenster	2	2	
	Papierkörbe	1	1	
	Lampen	2	2	
2/6	Schubladen	0		Wand Boden
Detail-form	Aktenablagen (Kunststoff)	0		linke Wand + Boden alles
	Ordner	0		
	Kartons	0		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	Bücher	0		
	Briefe	0		
	Zeitschriften	0		
	lose Blätter	0		
	Notizen	0		
	geheftete Stapel	0		
	Kulis / Bleistifte / Stabilos	0		
	Marker / Eddings	0		
	andere Stifte	0		
	Boardmarker	0		
	Zuleitungen	0		
	Datenkabel	0		
	CD / DVD / Speichermedien	0		
	Verlängerungen + Steckdo- senleisten	0		
	Taschen	0		
+	Pinn, Pinnwand	15	1	
2.1/6	Ölbilder	3		
Zusatz- form	Kalenderposter	1		
	Digicam, Olympus	1	1	
	Koffer, Digicam	1		
	Gitarrenständer	1		
	E-Gitarre	1		
	Gitarrenvorverstärker	1		
	Kabel, E-Gitarre	1		
	Buch, groß	1		
	Notiz	1		
	Gitarrenstimmgerät	1		
	Poster, klein	1		
	Pinnwand	1	1	
	Notiz	37		
	Visitenkarte	9		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	Vorhang	1		
3/6	Schubladen	0		Schrank
Detail- form	Aktenablagen (Kunststoff)	8	8	Eingang links
	Ordner	20	20	
	Kartons	18	18	
	Bücher	116	116	
	Briefe	0		
	Zeitschriften	0		
	lose Blätter	11	11	
	Notizen	0		
	geheftete Stapel	0		
	Kulis / Bleistifte / Stabilos	0		
	Marker / Eddings	0		
	andere Stifte	0		
	Boardmarker	0		
	Zuleitungen	0		
	Datenkabel	1	1	
	CD / DVD / Speichermedien	11	11	
	Verlängerungen + Steckdo- senleisten	0		
	Taschen	3	3	
+	Bewegungssensor	1		
3.1/6	Fackel	5		
Zusatz- form	Schaumstoffplatte	4		
	Hängeakte	8		
	Atari Spiel	4		
	Tastatur	2		
	Sprühdose, Lack	5		
	Message-Pad Apple	1		
	Kabel, seriell	1		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	PC Steckkarte	7		
4/6	Schubladen			PC Schreibtisch
Detail- form	Aktenablagen (Kunststoff)	1	1	Hauptschreibtisch, Fenster
	Ordner	0		
	Kartons	1	1	
	Bücher	2	2	
	Briefe	2	2	
	Zeitschriften	0		
	lose Blätter	13	13	
	Notizen	8	8	
	geheftete Stapel	0		
	Kulis / Bleistifte / Stabilos	6	6	
	Marker / Eddings	2	2	
	andere Stifte	3	3	
	Boardmarker	0		
	Zuleitungen	6	6	
	Datenkabel	21	21	
	CD / DVD / Speichermedien	3	3	
	Verlängerungen + Steckdo- senleisten	0		
	Taschen	1	1	
+	Plüschtier	1		
4.1/6	Tasse	1		
Zusatz- form	Passbild	1		
	Figur, Holz	1		
	Aspirin	1		
	Visitenkarte	150		
	Monitor	1	1	
	Tastatur	1	1	
	Tischwecker	1		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	Platine, Löt-	4		
	Figur, Elefant	1		
	Tischuhr	1	1	
	PC Lautsprecher	2	2	
	Visitenkarte	15		
	Block, Haftnotizen	1		
	Stifthalter	1	1	
	Messer	1		
	IC-Tube	1		
	Maus	1	1	
	Pad, Maus	1		
	Telefon	1	1	
	Ladestation	2		
	Batterie, AAA	1		
	Handytasche	1		
	Münze	22		
	Digicam	2	2	
	Handy	1		
	Flasche, PET	2		
	Orange, Frucht	1		
	Mini Disc	6		
	Particle	1		
	Handyakku	1		
	Wasserstoffbrennzelle	1		
	Display, Einbau	1		
	Hub	1		
	Card Reader	1		
	TetraPack Beutel, Orange	1		
	Hub, USB	1		
	PC, Mini-Tower	1	1	
	Reisszwecke, Pckg.	1		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
5/6	Schubladen	4	4	Tische Mitte
Detail- form	Aktenablagen (Kunststoff)	7	7	Tische Mitte, klein und groß
	Ordner	0		
	Kartons	6	6	
	Bücher	9	9	
	Briefe	42	42	
	Zeitschriften	2	2	
	lose Blätter	20	20	
	Notizen	0		
	geheftete Stapel	11	11	
	Kulis / Bleistifte / Stabilos	0		
	Marker / Eddings	0		
	andere Stifte	3	3	
	Boardmarker	0		
	Zuleitungen	1	1	
	Datenkabel	5	5	
	CD / DVD / Speichermedien	3	3	
	Verlängerungen + Steckdo- senleisten	5	5	
	Taschen	8	8	
+	Switch	1		
	Flasche, PET	1		
5.1/6	Wein	1		
Zusatz- form	Tasse	1	1	
	Block, DinA5	1		
	Aufhängehaken	12		
	Terminplaner	1		
	Kopfhörer, Mikrofon	2	2	
	Geodreieck	1		
	Schlüsselanhänger	3		
	Ausweis	7		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	Metalldose	1		
	Tesarolle	1	1	
	Teelicht	1		
	Platine, Löt-	2		
	Handyantenne	1		
	Visitenkarte	100		
	externe HD, DVD	3		
	Atari Spiel	1		
	Werkzeugkoffer	2	2	
	PIC Prozessor	1		
	Büroklammer	1		
	Notebook DVD Laufwerk	1		
	Notebook Akku	1		
	Sandalen	1		
	Netzteil	5		
	Mütze	3		
	Tischroboter Spielzeug	1		
	Diktiergerät	1		
6/6	Schubladen	0		Rollcontainer
Detail- form	Aktenablagen (Kunststoff)	0		
	Ordner	0		
	Kartons	6	6	
	Bücher	0		
	Briefe	9	9	
	Zeitschriften	11	11	
	lose Blätter	0		
	Notizen	0		
	geheftete Stapel	1	1	
	Kulis / Bleistifte / Stabilos	25	25	
	Marker / Eddings	10	10	
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	andere Stifte	10	10	
	Boardmarker	0		
	Zuleitungen	0		
	Datenkabel	9	9	
	CD / DVD / Speichermedien	31	31	
	Verlängerungen + Steckdo- senleisten	4	4	
	Taschen	1	1	
+	Gameboy Hüllen Pckg.	3		
+	Visitenkarte	55		
6.1/6	Akku	1		
Zusatz- form	Knopfzelle	1		
	Steckeradapter	4		
	Gehäuse, klein	3		
	Notebook HD	1		
	Platine, Löt-	8		
	Münze	11		
	Spielekonsole	2		
	Netzteil	5		
	Block, Haftnotizen	2		
	Büroklammer	1		
	Schrauben	62		
	Tesarolle	2		
	Radiergummi	2		
	Reisszwecke, Pckg.	1		
	Schlüssel	6		
	Schere	1	1	
	Klebestift	3		
	Messer	2		
	Flasche, Glas	1		
	Spindel, DVD	2		
	Diskettenbox	1		
Fortsetzung nächste Seite ...				

Fortsetzung ...				
	Sticker	4		
	Kaugummi	4		
	Zahnbürste	1		
	Bleispitzer	1		
	Messingrohr	1		
	Glühbirne	1		
	Lüfter	1		
	Abdeckung	4		
	Laptop	1		
	Maßstab	1		
	CD-Hülle	1		
	Overheadfolie Pckg.	1		
	Lautsprecher	1		
	Kleinteile	35		
	Maus	1	1	
	IC-Tube	1		
	Solarzelle	1		
	Landkarte	1		
	Mainboard	1		
	Gesamt / Relevant	1274	538	